Nonlinear Fault Detection, Isolation and Recovery Techniques for Unmanned Systems

FINAL REPORT

Solicitation No: W7701-052053/A
Contract No. W7701-052053/001/QCL
File No. QCB-5-20386

Prepared by:
K. Khorasani, Ph.D., P. Eng.
Department of Electrical and Computer Engineering
Concordia University
Montreal, Québec H3G 1M8

Submitted to:
Dr. C. A. Rabbath
Precision Weapons
Defence R&D Canada – Valcartier
2459 Pie-XI Blvd. North
Quebec City, Quebec G3J 1 X5

DRDC-VALCARTIER-CR-2007-295

March 30, 2007

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **30 MAR 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **Nonlinear Fault Detection, Isolation and Recovery Techniques for Unmanned Systems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Concordia University,Department of Electrical and Computer Engineering,1514 St. Catherine West,Montreal, Quebec Canada H4G 2W1,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**237**	

EXECUTIVE SUMMARY

To warrant satisfactory unmanned aerial vehicle flight and performance, despite the presence of actuator/engine faults and failures, which are bound to occur in adversarial and complex environments, there is a need to develop fault/failure detection and recovery systems. This is especially critical to ensure mission success, and more importantly, safety of civilian and military personnel. This report investigates several techniques of fault/failure detection and recovery suitable for small-scale unmanned aerial vehicles.

SOMMAIRE

Pour garantir le vol et la performance de véhicules drones, malgré la présence de fautes et de pannes des actionneurs et moteurs, qui peuvent survenir lors de missions dans des environnements complexes, il est nécessaire de développer des systèmes de détection de fautes/pannes et de recouvrement. Cela est essentiel pour assurer le succès de la mission, et revêt une très grande importance pour la sécurité des civils et des militaires. Ce rapport étudie différentes techniques de détection de fautes/pannes et de recouvrement qui sont applicables aux drone de petites dimensions.

ABSTRACT

In order to avoid adverse consequences due to failures, it is desirable to have an advanced failure detection and isolation (FDI) system that detects and identifies anomalies early to minimize the damage, and that can remedy as many failures as possible. In complex systems, fault diagnosis is typically accomplished using a hierarchical approach. In our proposed autonomous unmanned vehicle (UAV) system, fault diagnosis, isolation and recovery (FDIR) is accomplished by using a hierarchical and decentralized approach. At this level of the hierarchy the model based or analytical redundancy based approach to FDIR would require a mathematical model of the process or sub-process under consideration. Based on this knowledge quantities called residuals will be generated. The residuals should be small or close to zero when there are no failures in the system. On the other hand, they should become nonzero and grow large if there are malfunctions in the system. This will accomplish the failure detection. The next important task will be the design of a fault isolation module that would isolate the faulty components or subsystems. There are two major approaches to the design and implementation of recovery procedures; One is to synthesize the procedures for every possible failure mode at the design stage. Once the diagnostic and recovery system is activated, it monitors the system and if it detects a failure, then the system will initiate the appropriate recovery procedure. In the other approach, suitable recovery procedures are generated "on-line" upon the detection of failures. In this report, we will examine the advantages and drawbacks of the above approaches in our framework.

RESUME

Pour éviter les conséquences néfastes dues aux fautes, il est souhaitable de développer un système de détection et d'isolation de pannes qui détecte et identifie les anomalies suffisamment rapidement pour minimiser les dommages. Le diagnostique des fautes est généralement réalisé par le biais d'une approche hiérarchique pour des systèmes complexes. Pour le système de drone proposé, le diagnostique, l'isolation et le recouvrement (FDIR) repose sur une approche décentralisée et hiérarchique. L'approche analytique de FDIR requiert un modèle mathématique du drone. Le modèle utilisé permet de générer les résiduels. D'une part, les résiduels ont une valeur proche de zéro lorsque le système est exempt de panne. D'autre part, les résiduels ont une magnitude non-négligeable lorsque des anomalies se présentent au sein du système. La génération des résiduels permet de détecter les pannes de façon analytique. Une autre composante importante du système FDIR est le module d'isolation de la faute ou de la panne. Il y a deux approches pour la conception et l'implantation de procédures de recouvrement. Une propose d'inclure tous les modes de fautes et de pannes possibles dans le design. L'autre propose de générer en ligne une procédure de recouvrement une fois la panne détectée. Dans ce rapport, nous allons examiner les avantages et les inconvénients des approches ci-haut mentionnées.

Nonlinear Fault Detection, Isolation and Recovery Techniques for Unmanned Systems

FINAL REPORT

Solicitation No: W7701-052053/A
Contract No. W7701-052053/001/QCL
File No. QCB-5-20386

Prepared by:
K. Khorasani, Ph.D., P. Eng.
Department of Electrical and Computer Engineering
Concordia University
Montreal, Québec H3G 1M8

Submitted to:
Dr. C. A. Rabbath
Precision Weapons
Defence R&D Canada – Valcartier
2459 Pie-XI Blvd. North
Quebec City, Quebec G3J 1 X5

March 30, 2007

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) are indeed among the most complicated systems that are being developed. Since the motion of UAVs is specified in a three dimensional space and they operate in an environment subjected to a high degree of uncertainties and disturbances, the problem of precise and accurate control of such vehicles is difficult and requires complicated control systems theory. The importance of utilizing UAVs in a wide variety of civilian and military missions has been recognized by many funding and research agencies, as well as by a large number of companies in the private sector. On the other hand, with an increasing requirement for UAV control systems to be more secure and reliable, fault tolerance in such control systems is becoming more and more critical and important.

In the presence of undesirable effects such as failures in the actuators, control surfaces, or sensors, the UAV control systems must be responsive and adaptive to such failures. Under these circumstances there may be a necessity to adjust the control laws to recover the UAV from the effects of anomalies and failures. Furthermore, the required adjustments of control laws must be done expeditiously in a relatively short period of time to guarantee the operation of the safety critical subsystems of the UAV. Specifically, it is required to develop an autonomous fault diagnosis and reconfigurable control systems.

In particular for multiple UAVs operating within a given formation, if any of the vehicles undergoes a failure and the UAV controller is not equipped with autonomous fault tolerant capabilities, the stability and formation of the UAVs may not be maintainable and could lead to instability. Moreover, in leader-follower formation flight schemes, the failure of the leader UAV, in which the formation controller resides, may result in the total loss of performance for all the other follower UAVs. Therefore, the fault tolerant control systems for multiple UAVs environment needs to be fully investigated.

Advanced computers have resulted in more capable and advanced UAV flight systems. Thus, activities like navigation and maneuver planning, command planning and sequencing, and fault diagnosis and recovery can all potentially be autonomously handled onboard the UAV. Aside from the obvious cost savings realized by smaller operations staff, there are traditional advantages to placing some of these functions on the UAV. For example, onboard fault diagnosis and recovery algorithms can detect, identify, and remedy UAV faults, both minor and major, *in real-time*, possibly saving the mission in the process. Moreover, the ability to plan UAV activities onboard allows the UAV to respond to major instrument failures or other anomalies without impacting the remaining healthy UAV subsystems. In the event of a major anomaly, the UAV can respond quickly and generate a new sequence of commands to carry out the remaining possible mission objectives. For example, after the occurrence of a fault in the leader UAV instruments, it can go to the safe mode of operation, if full recovery is not feasible, and the leader responsibilities will be allocated to some other UAV in the formation.

Conventionally, fault diagnosis in fault tolerant control systems are achieved and ensured through hardware redundancy, that is, by including redundant actuators and sensors in the system (UAV in this project). The control and measurement channels are generally made duplicated or triplicated in hardware. The main disadvantage of physical redundancy is the additional cost and the corresponding increase in complexity of operation. Moreover, the weight of the UAV and the maintenance requirements will subsequently increase. Consequently, analytical redundancy approach, which makes use of the mathematical model of the system and relationships between sensor output and actuator inputs (control signals), has been proposed and are increasingly being employed in fault tolerant control systems.

Building on methods presently available in several fields including estimation and observation theory, system identification, robust and adaptive control, intelligent systems, and system health monitoring, novel and innovative techniques have been proposed and fully investigated in this project by developing new capabilities and by relaxing/removing the limitations of the current state-of-the-art technology in diagnosis, isolation and recovery as they relate to autonomous aerial vehicles.

Our research was mainly motivated by the fact that most available fault detection, isolation and recovery (FDIR) techniques lack the capability to handle the following two practical issues:

(i) Incomplete and varying knowledge concerning fault modes

(ii) Existence of inevitable modeling and enviromental uncertainties such as un-modeled dynamics, parameter uncertainties and disturbances.

To address the above two issues, we have integrated the analytical techniques with the knowledge-based/intelligent methodologies. By taking advantage of the efficient, formalized and mathematically proven capabilities of model-based approaches, and the capability of knowledge-based/intelligent systems in handling incomplete, less formalized and even uncertain knowledge, an integrated and hybrid framework is developed to produce more flexible and practical solutions.

1.1 Objective of the Project

In order to avoid adverse consequences due to failures, it is desirable to have an advanced failure detection and isolation (FDI) system that detects and identifies anomalies early to minimize the damage, and that can remedy as many failures as possible. In complex systems, fault diagnosis id typically accomplished using a hierarchical approach. In our proposed autonomous unmanned vehicle (UAV) system, fault diagnosis, isolation and recovery (FDIR) is accomplished by using a hierarchical and decentralized approach. At this level of the hierarchy the model based or analytical redundancy based approach to FDIR would require a mathematical model of the process or sub-process under consideration. Based on this knowledge quantities called *residuals* will be generated. The residuals should be small or close to zero when there are no failure in the system. On the other hand, they should become nonzero and grow large if there are malfunctions in the system. This will accomplish the failure detection. The next important task will be the design of a fault isolation module that would isolate the faulty components or subsystems.

There are essentially three distinct FDIR approaches that have been investigated in this project:

- Techniques that use physically based model of the unmanned vehicles, and use the model based techniques from estimation and control domain to tackle the problem.
- Techniques that are artificial neural networks(ANN) for model development and use those models along with other intelligent networks for fault detection and isolation purposes.
- Finally, use the combination of the above two methodologies for both model development as well as the fault detection, isolation and recovery logic.

There are two major approaches to the design and implementation of recovery procedures, One is to synthesize the procedures for every possible failure mode at the design stage. One the diagnostic and recovery system is activated, it monitors the system and if it detects a failure, then the system will initiate the appropriate recovery procedure. In the other approach, suitable recovery procedures are generated "on-line" upon the detection of failures. We will examine the advantages and drawbacks of the above approaches in our framework.

Chapter 2

Issues of Cooperative UAV Control

In a number of future civilian and military operations it is expected that one needs to utilize a large number of unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), and unattended ground sensors (UGSs) that are configured to concurrently perform Command, Control, Communication, Computing, Intelligence, Surveillance, and Reconnaissance (C4ISR) offering extended capabilities. The expected sheer number of unmanned assets or nodes (this is UGVs, UAVs, and UGSs) that are planned to be used and their corresponding dynamics, coupling interactions, network bandwidth constraints, data latency (delays in data time of arrival), scheduling and real-time requirements, data acquisition, priorities, processing and routing requirements all to be controlled and managed seamlessly, would introduce challenging problems for the operators and managers. These challenges call for the development of innovative and novel autonomous, intelligent and distributed algorithms where a cluster of unmanned systems (UMS) in collaboration and cooperation with other UMS clusters would autonomously or semi-autonomously achieve and contribute to accomplishing the desired performance specifications and requirements with minimal support and involvement of commanders and operators.

Specifically, typical military missions of multiple UAVs involve 3-D stereographic mapping, platforms for wide aperture sensors, grouping and regrouping for purposes of attack, damage assessment, decoys and high-G-force maneuvers. In civilian applications, UAV teams are used in the areas such as coastal and border surveillance, embargo enforcement, humanitarian and disaster assistance, bridge inspection, fire-spotting, border patrol, and oceanographic and meteorological data collection [6].

Current issues in cooperative UAV control are presented by topics [7]. Aerial surveillance and tracking allows vision-based control. Collision and obstacle avoidance and formation reconfiguration ensure safe cooperation for groups of aircraft. High level control is needed to coordinate large groups of UAVs and for real-time human interfacing, and all of these depend on system hardware and communications.

2.1 Aerial Surveillance and Tracking

Aerial Surveillance and Tracking: Surveillance, detection and tracking have been studied and developed in the past for static security cameras, ground vehicles and manned aircraft. The main challenges of migration to UAVs are due to the physical constraints of the platform and the strict real-time requirements the applications. It is usually not feasible to carry significant amounts of hardware due to size, weight, and power limitations. At the same time it is of paramount importance to process the sensor input at a real-time rate: while some seconds of delay in the recognition of an intruder for a security camera application may not be a problem, this could result in loss of the target in a UAV application due to the high speed of the aircraft [7].

2.2 Path Planning

Path planning is one of major issues that need to be addressed in the development of multiple UAVs control. A path planning algorithm computes a trajectory from the UAV's present location to a desired future location, e.g. a target [8].

A good path planning algorithm for UAVs must possess several important attributes, making its design a multiple-objective optimization problem [8]. First it must compute a stealthy path, steering the aircraft and its radar signature around known enemy radar locations. This is difficult because no aircraft scatters or reflects radar radiation uniformly in all directions. Rather, radiation is radiated more strongly in some direction, and less strongly in others. Second, generated trajectories should be of minimal length, subject to the stealthy constraint, and also satisfy the aircraft's dynamic constraints. Third, the path-planning algorithm must be compatible with the cooperative nature envisioned for the UAV. A typical mission might involve multiple UAVs attacking single, well-defended target. The path planner would be a component – a subroutine – of the hybrid control system that ensures all UAVs arrive simultaneously. And finally, path planner algorithms are expected to be coded in software that runs on an airborne processor. Thus, they must be computationally efficient and "real-time", enabling the UAV to re-plan its trajectory should an unforeseen threat arise.

Towards these objectives, a number of studies have conducted [9, 10, 11, 12, 13, 14]. For example, Jun and D'Andrea [15] proposed a path method for UAVs by using a probability map. The approach in this paper is similar to those in [8] and [16] in which one decomposes the problem into two steps: first the generation of a preliminary polygonal path by using a graph, and then a refinement of the path. In [15], the nodes and links of the graph are based directly on the probability map. The authors also consider the effects of moving threats, changes in the probability map, and multiple vehicles, and perform an analysis of the effects of refinement on the initial path.

2.3 Collision and Obstacle Avoidance

Collision and obstacle avoidance are the basis of safe UAV flight. Here, collision avoidance refers to non-collision among cooperative UAVs, and requires that aircraft either detect each other using sensors or report their positions [7]. Collision avoidance is usually incorporated into a cooperative flight algorithm. Obstacle avoidance is distinguished by lack of knowledge and cooperation: an obstacle may take any form and may be either inert or hostile (including seeking a collision), hence detection is a primary problem. The ability to see and avoid obstacles is a necessary condition for flight in civil airspace, and for low altitude applications such as urban canyon navigation. Once an obstacle has been detected, the flight path must be altered in order to ensure aircraft safety while minimizing deviation from the optimal path and continuing to ensure collision avoidance [7].

2.4 Cooperative Control

2.4.1 Cooperative Control Law

In order to achieve the cooperative control of multiple UAVs, the high level cooperative control law has to be applied to the UAV team. As discussed previously, typical missions of multiple UAVs include 3-D stereographic mapping, platforms for wide aperture sensors, coastal and border surveillance, etc. Towards these ends, different cooperative control laws need to be developed. For example, some of missions require the UAVs flying in formation, in which the dynamics of the UAVs is tightly coupled either by control laws governing the formation behavior or by a combination of aerodynamics formation control law [17].

Leader-Follower approach is a typical formation flight control law, in which the follower UAV follows the trajectory of the leader, taking the other aircraft as reference to keep its own position inside the formation [18]. To compute the distance to its reference, each aircraft acquires its position $P = (X, Y, H)$ from a GPS-based position sensing system and receives, through appropriate communication channels, the other aircraft's positions $P_R = (X_R, Y_R, H_R)$. The formation control is also responsible for having each aircraft follow a prescribed path. The formation control receives path information in terms of velocity, heading, and altitude $T_R = (V_R, H_R, \Psi_R)$. Then the received data vector becomes $R_R = (X_R, Y_R, H_R, V_R, \Psi_R)$. The commanded trajectory for the "inner-loop" control is $T_C = T_R + \Delta T$.

In addition, there are two types of scenarios can be implemented:

- Leader Mode: Both follower1 and follower2 take the trajectory references from the Leader of the formation.
- Front Mode: Each aircraft takes its reference from the preceding one. In this case, follower1 is referred to the leader and follower2 is referred to follower1.

In addition, reference [19, 20, 21] considers the control of interacting subsystems whose dynamics and constraints are uncoupled, but whose state vectors are coupled non-separably in a single centralized cost function of a finite horizon optimal control problem. For a given centralized cost structure, the authors generate distributed optimal control problems for each subsystem and establish that the distributed receding horizon implementation is asymptotically stabilizing. The communication requirements between subsystems with coupling in the cost function are that each subsystem obtains in the previous optimal control trajectory of those subsystems at each receding horizon update.

Furthermore, Olfati-Saber and Murray [22] provided a unified graph-theoretical framework that allows ones to formally define formations of multiple vehicles and their stabilization issues. They clarified the important role of graph rigidity and minimally rigid graphs in construction of structural potential functions and manipulation of multiple formations. This includes formal representation of split, rejoin, and reconfiguration maneuvers for formations of multi-vehicle systems.

Other than formation flight, some of the missions require the control of spatially distributed UAV systems that are not coupled dynamically by aerodynamic forces or low-level control law. For example, in certain scenarios, the objectives are for the UAVs to arrive at their targets simultaneously (UAV rendezvous) with sufficient fuel to return to base while minimizing the risk of detection by threats [17]. From a strategic standpoint, simultaneous arrival at the targets increases the element of surprise and decrease the risk of detection. Once the targets have been hit, the cooperative control must plan the egress of the UAVs.

Mclain et al. [17] proposed a cooperative control law to approach the cooperative control UAV rendezvous. In their work, firstly, the path planner for each UAV must produce a family of paths from its initial location to its target. The cost for each path in the family of paths is based on the proximity of the each path to the threats. Moreover, a decomposition solution is developed in order to not only find the best solution for the team as a whole, but also compute the team optimal solutions in a decentralized manner.

Another kind of cooperative UAV control involves a large number of UAVs that can be considered as a flock. Flocking behavior is based on prioritized interactions between three agent types. When a flock member (α) detects an obstacle, it creates virtual β and γ agents, and its behavior is then determined by the location of all agents, prioritized to ensure obstacle avoidance [5].

Interactions between α agents are represented using a structural net and can be characterized as attempting to keep a specified neighbor separation. The control law is based on the structural energy of the flock. A development of interest is the representation of Reynolds' three boid protocol using a single protocol for $\alpha - \alpha$ interaction [7].

2.4.2 Cooperative Reconfigurations

When UAVs perform a cooperative task by flying as a group, they may be considered flying in a formation or a flock. Groups must safely reconfigure in response to changing missions, UAV populations, environments, and UAV faults. The approaches of formation reconfiguration are related to the high level control very much.

Using a hybrid systems approach, a specific formation can be seen as a finite state machine, where transitions between different formation (or state machines) can be triggered by changes in mission or environment. The transition processing must reassign aircraft to positions in the new formation and provide trajectories from the initial formation positions to the new ones.

In addition, for a loosely defined cooperative UAV control such as a flock, reconfiguration may be inherent in the formation flight algorithm. Specifically, the framework proposed in reference [22, 23]enables ones to address split/rejoin and squeezing maneuvers for nets/flocks of dynamic agents that communicate with each other.

2.5 Cooperative Control Management (Supervisor)

In order to achieve the high level cooperative control, management of the aircraft team has to be addressed. Management of the aircraft team can be *centralized* or *decentralized* [18]. In the former case, one team manger acts as a supervisor for all the aircraft and manages the topology of the channels used to exchange information among them. In a decentralized management scheme, one the other hand, each aircraft is given a certain level of decision capability, while the whole formation must be capable of reconfiguring, making decisions, and achieving mission goals. The problem with distributed management is that: the distributed decision-making algorithm must produce deterministic results on all the managed components; that is, on all the aircraft, conflicting decisions must be avoided for formation safety. The advantages of distributing the management are several:

- Only inter-aircraft information must be exchanged.
- The same data channels used by formation-keeping control could be used to exchange management information.
- Very low power or, alternatively, non-radio-based communications such as optical sensors can be used because of the very small distances among aircraft. This could be very important for military applications. (as opposed to aircraft-ground communications).
- Reaction times can be minimized. (as opposed to aircraft-ground communications).

2.6 Hardware and Communications

Sensor choices are intimately tied to the platform and constraints on payload size, weight, and power. Vision systems utilizing conventional and/or infrared cameras are commonly used for mapping and obstacle detection/avoidance applications. One of the major issues for collaborative unmanned aircraft is wireless communication with other cooperating aircraft and/or the ground. Aircraft-to-aircraft communication is a more recent problem with the increasing availability of wireless LAN and GSM technologies. High-bandwidth communications may be needed. Aircraft-to-aircraft communications may also be required in out of line-of-sight conditions due to distance or obstacles.

More importantly, the communications graph should be redundant from the standpoint of the capability of propagating information. In the event of faults, this redundancy leaves room for reconfiguration.

Optimization of Communication Channels

To use an optimization technique, each arc must be given a weight. The cost function will minimize the total cost of the information paths throughout the formation using the arcs' weights to evaluate the cost of a connection.

Redundant Channels

Once an optimal non-redundant set of arcs has been found, lower priority suboptimal redundant arcs could be added to the active channels for two reasons: for mixing sensor information for control and as alternatives in emergency and fault conditions.

Graph Theory Approach to Optimization

Under certain assumptions proposed by [18], the optimization problem can be configured as a shortest path problem using Dijkstra algorithm [24].

2.7 Fault Detection, Isolation and Recovery (FDIR) for UAVs and Potential Research Areas of this Project

Along these lines, UAVs are indeed among the most complicated systems that are being developed. Since the motion of an aircraft is specified in a three-dimensional space and aircraft operate in an environment subjected to a high degree of uncertainties and disturbances, the problem of precise and accurate control of such vehicles is difficult and requires complicated control system theory. With an increasing requirement for aircraft flight control systems to be more secure and more reliable, fault tolerance in such control systems is becoming more and more critical and important.

In the presence of undesirable effects such as faults in the actuators, control surfaces, or sensors, the aircraft flight control systems must be responsive and adaptive to such faults. Under these circumstances there may be a necessary to adjust and fine-tune the control laws to recover the aircraft from the effects of anomalies and faults. Furthermore, the required adjustments of control laws must be done expeditiously in a relatively short period of time to guarantee the operation of the safely critical subsystems of the aircraft. Specifically, what is required is to develop an autonomous fault diagnosis and reconfigurable flight control systems.

Based on the recognized issues in cooperative UAV control, Figure 2.1 depicts the research areas we will investigate in this project. Due to limited time and resources, we are not going to investigate areas like Mission Planning, Path Planning, Collision Avoidance, and Obstacle Avoidance.

2.8 UAVs' Cooperative Tasks

A cooperative system is loosely defined as a collection of dynamical entities that share information to accomplish a common, though perhaps not singular, objective [25]. In order to refine the above definition, one has to define some fairly specific goals for cooperating UAVs. However, in order to define these goals, some assumptions should be devised. A fundamental assumptions is that the UAVs will be *multi-role,* that is, they each have multiple functions or capabilities such as searching for targets, destroying targets, reporting on target damage, various surveillance roles, a communication relay role, egress and landing. Also, good measures of effectiveness (MOEs) for a team of UAVs are often *task oriented* and likely multi criteria in nature. A task-based viewpoint portrays the UAVs as a constrained resource with multiple conflicting demands. Meanwhile, the UAVs are expected to do well on non task-based *performance* measures, such as increased flight time through the use of efficient trajectories and collision avoidance.

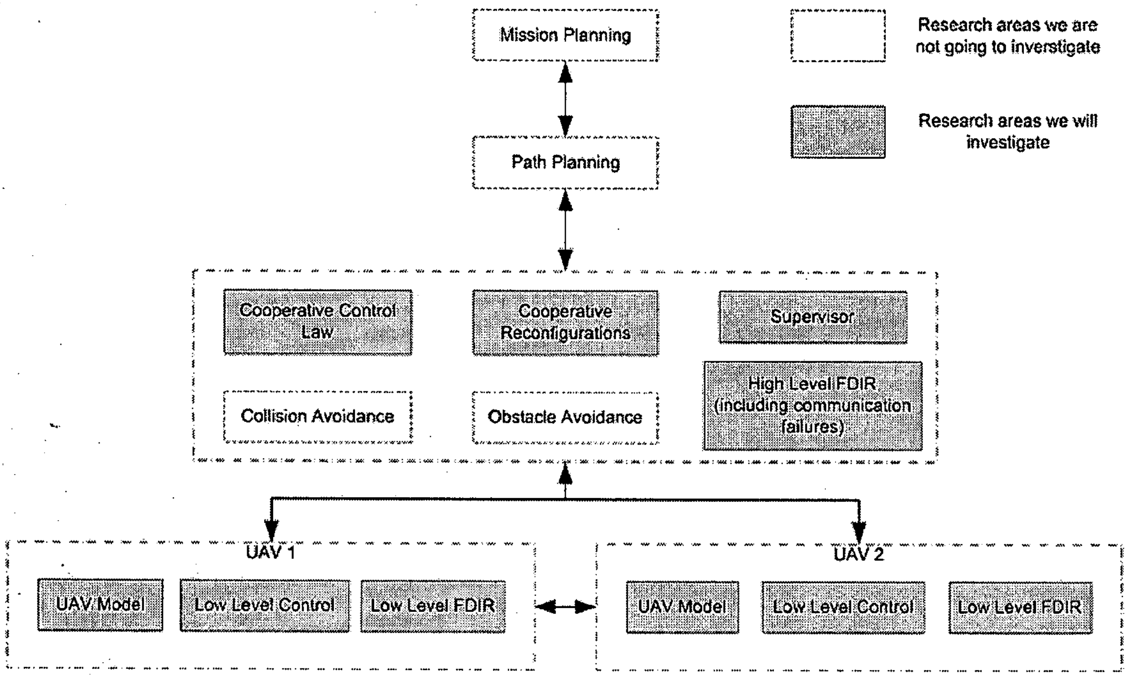


Figure 2.1: Potential Research Areas of this Project

In summary, cooperative systems are multi-role and simultaneously task-based, and performance-based. Task-based measures tend to evolve discretely, for instance, switching through modes of operation or multiple dynamic models. So for modeling cooperative systems, we require hybrid dynamics model with the discrete dynamics associated with all the task-based cooperative control functions while the continuous dynamics are associated with the performance-based and autonomous control functions. In such architecture, the operating mode and waypoints from the discrete controller are inputs to the continuous controller while the local measurements and measurements transmitted from other UAVs are inputs to the discrete controller. It should be noted that the discrete dynamics are much slower than the continuous ones.

Effective cooperation often requires that individual UAVs coordinate their actions [26]. Coordination can take many forms ranging from "staying out of each others way" to directly assisting another UAV. Basically, there exist two sorts of coordination among UAVs: (1) *global coordination,* in which a UAV coordinates its action with *every* other UAV in the team, and (2) *local coordination,* which is in contrast to the global one. However, due to communication constraints and computational feasibility, researchers are primarily interested in group coordination problems where the coordination takes place locally. Thus, one of the challenges in UAV cooperative control would be the design of local coordination strategies that will result in group cooperation. However, as was mentioned above we need to specify the cooperative task and its associated objective in order to have a good measure of effectiveness for assessing the performance of the applied coordination algorithm. In the subsequent sections we will briefly describe the existing cooperative tasks and missions together with their associated objectives reported in the literature.

2.8.1 Aerial Surveillance

There are numerous applications requiring aerial surveillance [26].

- Civilian Applications: monitoring forest fires, oil fields, and tracking wildlife.
- Homeland Security: border patrol and monitoring the perimeter of nuclear power plants
- Military Applications

Beard et al. [26] have proposed a cooperative control strategy for aerial surveillance that has been successfully flight tested on small UAVs. Their approach can be summarized in four steps:

(i) **Cooperation Objectives and Constraints:** the definition of a cooperation constraint and cooperation objective. Certain relationships between state variables of the UAVs are called cooperation constraints.

(ii) **Coordination Variable and Coordination Function:** the definition of a coordination variable as the minimal amount of information needed to satisfy the cooperation objective. Also, the relationship between the coordination variable and the cooperation objective known as the coordination function must be specified.

(iii) **Centralized Cooperation Scheme:** the design of a centralized cooperation strategy to minimize the team objective function assuming that each UAV has global knowledge of the coordination variable and the coordination functions of other UAVs in the team.

(iv) **Consensus Building:** the use of consensus schemes to transform the centralized strategy into a decentralized algorithm due to the noisy and non-persistent communication links and dynamically changing communication topology. Such an algorithm assures that each UAV in the team has consistent coordination information despite the inadequacies of the communication network.

As an example of aerial surveillance, Beard et al. [27] have investigated the cooperative forest fire surveillance problem extensively. In this problem, multiple UAVs are distributed around the perimeter of a growing forest fire. The main features of the proposed decentralized multiple-UAV approach include (1) the ability to monitor a changing fire perimeter, (2) the ability to systematically add and remove UAVs from the team (important for re-fueling), and (3) the ability to communicate time-critical information regarding the coordinates of the fire perimeter to fire fighters on the ground. However, it is assumed that the UAVs have a limited communication range that requires them to work cooperatively to relay fire perimeter information to the base station. In the proposed scenario, each UAV patrols a segment of the periphery and whenever it encounters another UAV, it exchanges information and reverses direction until it encounters another UAV or arrives at a predetermined rendezvous point. As mentioned above, it is also assumed that each UAV has limited fuel, which implies that it must periodically return to the base station for refueling.

2.8.2 Cooperative Attack

The goals of cooperative attack include enhancing UAV performance and increasing battlefield situational awareness. By sharing information on their own states and the state of their observation, UAVs may re-plan trajectories in-flight, determine what targets to attack, and increase the probability of target detection and classification [28].

2.8.3 Close Formation Flight

Early studies reported in [29] reported significant potential fuel savings that could be gained by close formation flight. Standard inner/outer loop designs are extended to close-formation flight in [30]. A behavioral approach to aircraft formation flight is given in [31]. The effects of communication constraints on close formation flight are studied in [18]. Rigorous conditions for stable formation flight with limited communication are developed in [32].

2.8.4 UAV Rendezvous Problem

Cooperative timing tasks for UAVs are of interest in many military missions. This problem is also known in the literature as UAV rendezvous problem [17] or target assignment and intercept for UAVs [33]. In the cooperative timing problem proposed in [33], the cooperation objective is defined as the simultaneous (or with a specified time spacing) arrival of UAVs at their targets (pre-assigned or dynamically assigned), while avoiding threats (known *a priori* or *pop-up*) and terrain-based obstacles (known a priori). The cooperation objective is achieved if the arrival time constraint is satisfied and the quality of the cooperation is measured by the extent to which threats and obstacles are avoided. From strategic point of view, simultaneous arrival at targets increases the element of surprise and decreases the risk of detection. In [17] the authors define the UAV rendezvous problem similarly with the extra constraint that the UAVs should eventually have sufficient fuel to return to base. In other words, once the targets have been hit, the cooperative control must plan the egress of the UAVs. In [33] the authors define the cooperative timing problem as a coordinated target assignment and intercept problem. It is defined as a scenario in which we have a group of MUAVs that are required to transition through N known target locations (static targets). However, in the region of interest there are a number of known and pop-up threats and it is desirable to have multiple UAVs arrive on the boundary of each target's radar detection region simultaneously. As an approach towards the solution of the problem, the authors have decomposed the scenario into several sub-problems given below:

- Assigning each UAV to a target such that each target has, if possible, multiple UAVs assigned to it with the preference of high-priority targets taken into account.
- For each team of UAVs assigned to a single target, an estimated time over target (TOT) should be determined that ensures simultaneous intercept and that is feasible for each UAV on the team.
- Determining a path (specified via waypoints) for each UAV such that it could complete the path in the specified TOT, given its velocity constraints.

- Transforming each waypoint path into a feasible trajectory in terms of turning rate and velocity constraints.

- Developing globally asymptotically stable controllers such that the UAVs track their specified trajectory.

The authors in [16] have addressed the problem of path planning and coordination of multiple UAVs in a coordinated UAV intercept or rendezvous problem. They define the cooperation objective as to jointly reach a target area while minimizing the combined flight's exposure to radar. The proposed approach is hierarchical in the sense that it consists of two levels. At the lower level is the path planning module which itself consists of a two step procedure. First, an approximation to the optimal path is made. Second, the initial path is refined to a flyable path by incorporating velocity and curving rate constraints. At the higher level, the coordination of the timing of the UAVs' arrival at the target area is performed by the coordination agent or rendezvous manager. In other words, the rendezvous manager calculates the Estimated Time of Arrival (ETA) as the coordination variable and communicates it to all the other UAVs in the team. The UAVs follow their planned trajectories, generated based on the calculated ETA, until an unplanned event, such as a pop-up threat occurs. Then the rendezvous manager determines a new ETA, based on the new information, and consequently a new trajectory will be generated.

2.8.5 Cooperative Target Search Problem

This problem is defined in [34] as the autonomous near-optimal trajectory generation in order for the UAVs to be able to cooperatively identify the maximum number of targets in an environment for which some a priori data about target distribution is available. More specifically, the objective of each UAV in the search problem is to move over the environment such that, at the end of the search, the maximum number of new targets has been identified. The a priori data about the environment is assumed to be given as information about the likelihood of targets being located in different regions of the environment. They have also assumed communication bandwidth constraints in their work. Towards the solution to the cooperative search problem, the authors have proposed an algorithm based on a discrete time stochastic modeling of the problem, which is then implemented with a dynamic programming technique.

The same authors as in [34] have considered in [35] and [36] the similar problem of cooperative UAV target search but this time in risky environments. In other words, they have considered the possibility of threats in the environment and these threats are explicitly represented in the model of the environment and in the planning algorithm. More specifically, the cooperation objective is refined to identify as many targets as possible and minimize the loss or damage of the UAVs during the mission.

2.8.6 Cooperative Target Search and Response Problem

In the cooperative search problem the main objective is the identification of the targets while in a search and response problem, other than target search the UAVs' are supposed to perform attack and damage assessment tasks on each target known or discovered through the *search* process. The authors in [37] have addressed the cooperative search and destroy problem. In their work, some target locations are suspected *a priori,* while the rest need to be detected gradually through search process. The tasks are determined in real-time by the actions of all UAVs and their sensory data, which makes the task dynamics stochastic. Thus, the tasks must be allocated to UAVs in real-time as they arise. While quick response is more important for known targets, efficient search is necessary to discover hidden targets. The authors show that there is always a trade-off between search and task response. As a consequence, they propose a hybrid algorithm which balances the search and task response.

Chapter 3

Definitions of UAV Faults

3.1 Component Faults

In this section, we will define component faults in UAV systems. Low level faults are those faults that can locally affect each UAV in the team. These faults can be classified as follows (Figure 3.1):

(i) Actuator Faults

(ii) Sensor Faults

(iii) Body Damage Faults

(iv) Transmitter/Receiver Faults

3.1.1 Actuator Faults

Common actuator faults in UAVs include [6]:

(i) Freezing or Lock in-place (LIP)

(ii) Float

(iii) Hard-over Fault (HOF)

(iv) Loss of effectiveness

In the case of LIP faults, the actuator "freezes" at a certain condition and does not respond to subsequent commands. HOF is characterized by the actuator moving to the upper or lower position limit regardless of the command. The speed of response is limited by the actuator rate limit. Float fault occurs when the actuator "floats" with zero moment and does not contribute to the control authority. Loss of effectiveness is characterized by lowering the actuator gain with respect to its nominal value. Different types of actuator faults can be parameterized as follows:

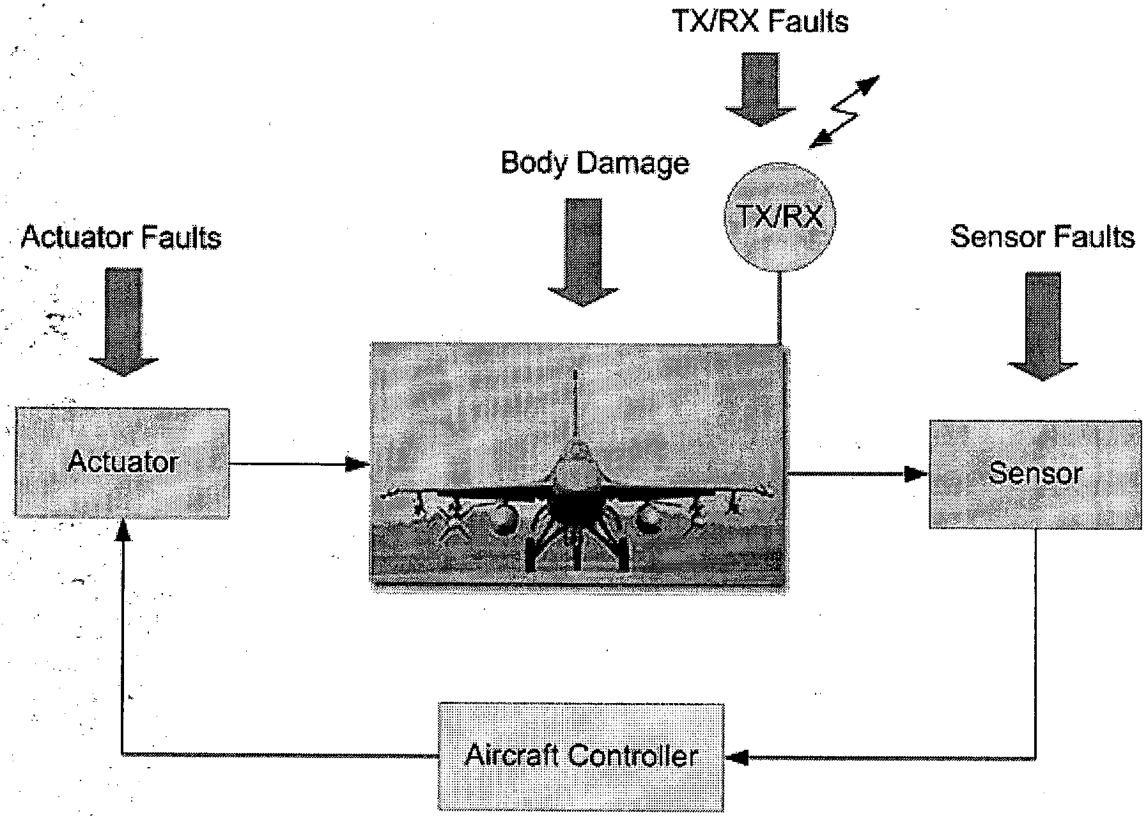


Figure 3.1: Low level faults in UAVs

$$u_{true} = \begin{cases} u_{cmd} & \text{(No Failure Case)} \\ k_i(t)u_{cmd} & 0 < \varepsilon_i \leq k_i(t) < 1, \forall t \geq t_F \text{ (Loss of Effectiveness)} \\ 0 & \forall t \geq t_F \text{ (Float)} \\ u_{cmd}(t_F) & \forall t \geq t_F \text{ (Lock-in-Place)} \\ u_{im} \text{ or } u_{iM} & \forall t \geq t_F \text{ (Hard-over)} \end{cases} \tag{3.1}$$

where t_F denotes the time instant of fault of the actuator , k_i denotes its effectiveness coefficient such that $k_i \in [\varepsilon_i, 1]$ and $\varepsilon_i$denotes its minimum effectiveness.

The following model

$$u_i = \sigma_i k_i u_{ci} + (1 - \sigma_i)\bar{u}_i \tag{3.2}$$

includes all above cases, where $u_i$is the actuator output, $u_{ci}$is the output of the controller (which at the same time, an input to the actuator), $\sigma_i = 1$ and $k_i = 1$ in no fault case, $\sigma_i = 1$ and $0 < k_i < 1$ in the case of loss of effectiveness fault and $\sigma_i = 0$ in other types of fault where $\bar{u}_i$ is a position at which the actuator locks in the case of float, lock-in-place and hard-over fault.

It is seen that the unknown parameters in the above model is the unknown actuator gain $k_i$in the case of loss of effectiveness, and the unknown position $\bar{u}_i$ at which actuator locks in the case of float, lock-in-place and hard-over faults.

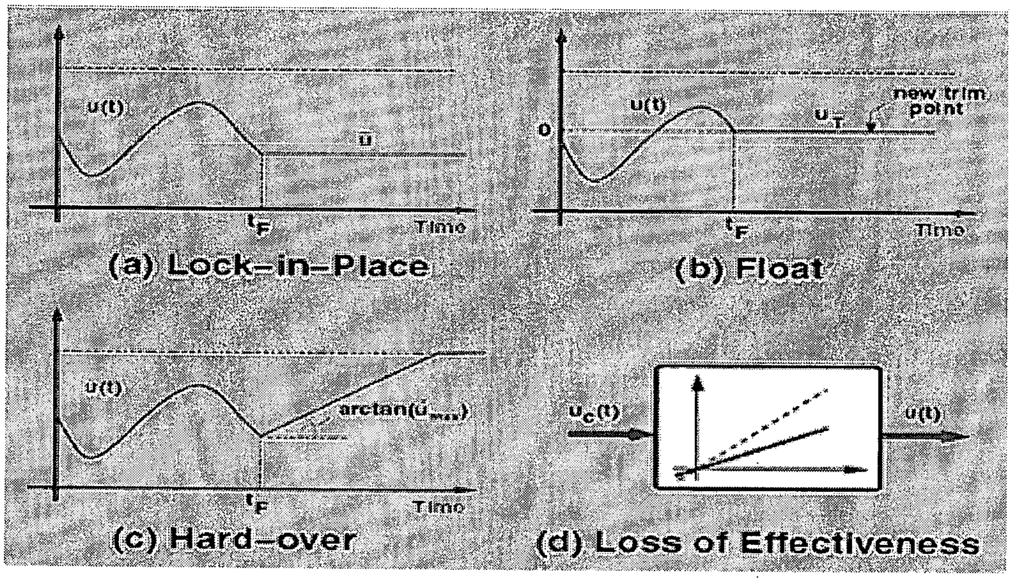


Figure 3.2: Actuator Faults [1]

3.1.2 Sensors Faults

Common sensor faults include [38]:

(i) Bias

(ii) Drift

(iii) Performance degradation (Loss of accuracy)

(iv) Freezing

Figure 3.3 shows the effect of above faults on the output of the sensor. Different type of sensor faults can be parameterized as follows:

$$y_i(t) = \begin{cases} x_i(t) & & \forall t \geq t_o & \text{No Failure case} \\ x_i(t) + b_i & \dot{b}_i(t) = 0, b_i(t_{Fi}) \neq 0 & & \text{Bias} \\ x_i(t) + b_i(t) & |b_i(t)| = c_i t, 0 < c_i << 1 & \forall t \geq t_{Fi} & \text{Drift} \\ x_i(t) + b_i(t) & |b_i(t)| \leq \bar{b}_i, \dot{b}_i(t) \in L^\infty & \forall t \geq t_{Fi} & \text{Loss of Accuracy} \\ x_i(t_{Fi}) & & \forall t \geq t_{Fi} & \text{Sensor Freezing} \\ k_i(t) x_i & 0 < \bar{k}_i \leq k_i(t) \leq 1 & \forall t \geq t_{Fi} & \text{Calibration Error} \end{cases} \tag{3.3}$$

where t_{Fi} denotes the time instant of fault of the ith sensor, and b_i denotes its accuracy coefficient such that $b_i \in [-\bar{b}_i, \bar{b}_i]$ where $\bar{b}_i > 0$. Also, it is seen that $k_i \in [\bar{k}_i, 1]$, where $\bar{k}_i > 0$ denotes some minimum sensor effectiveness.

We can conclude that the following model include all above cases, except the case freezing:

$$y = K_m x + b \tag{3.4}$$

where $K_m$is a positive definite diagonal matrix whose elements are slowly varying within $[\bar{k}_i, 1]$, and elements of vector $b$that slowly varying within $[-\bar{b}_i, \bar{b}_i]$.

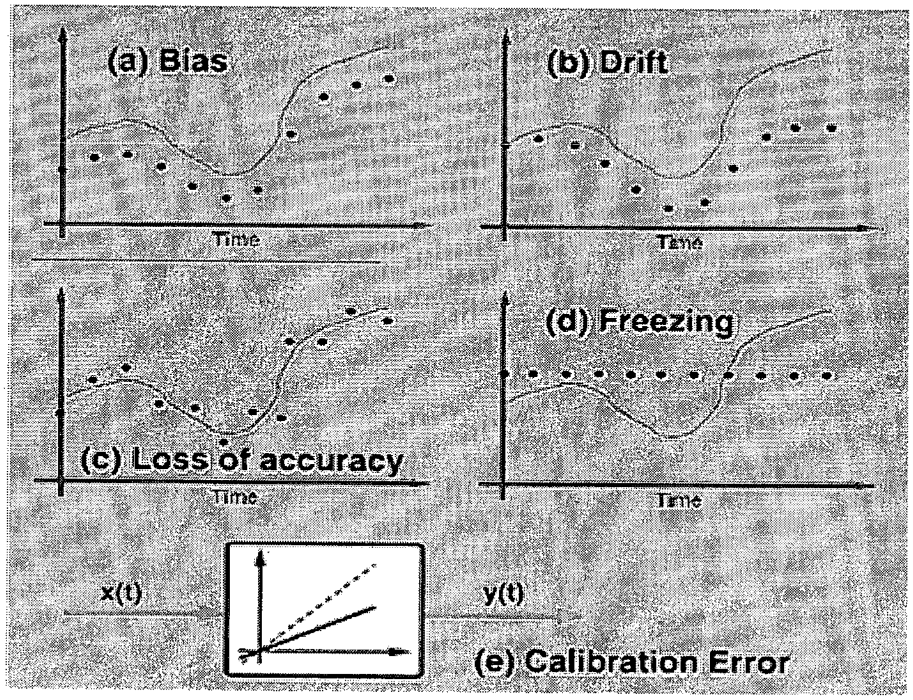


Figure 3.3: Sensors Faults [1]

3.1.3 Body Damage Faults

Body damage faults are those faults that are due to damage in the body of UAVs. Modeling of these types of fault is very difficult and needs numerous experiments. In this section we consider two body damage faults

- Wing Damage
- Control Effectors Damage

Wing Damage Faults

As discussed in Ref. [39] based on numerical modeling and numerous wind-tunnel experiments, researchers from Boeing developed Tailless Advanced Fighter Aircraft (TAFA) models for different damage conditions, including the case of no damage and the cases of increments of certain percentage of wing damage up to the 100% damage. For percentages of the damage in between the corresponding matrix norms, a polynomial approximation in the terms of the percentage-of-damage parameter $\rho \in [0, 1]$ and carried out interpolation that resulted in acceptable approximation with a fifth-order matrix polynomial in ρ. In this approach they consider a linearized model of a TAFA in the presence of wing damage. The generalized aircraft model is of the form

$$\dot{x}(t) = A_D(\rho)x(t) + B_D(\rho)u(t) \tag{3.5}$$

where $x \in R^n$ and $u \in R^m$ denote, respectively, the state and control input vectors. The time-varying nature of matrices A_D and B_D is caused by the wing damage, which affected the dynamics of the aircraft in an abrupt fashion. In [39] the following form of these matrices was proposed:

$$
\begin{aligned}
A_D(\rho(t)) &= A_{D0} + A_{D1}\rho(t) + A_{D2}\rho^2(t) + A_{D3}\rho^3(t) + A_{D4}\rho^4(t) + A_{D5}\rho^5(t) \\
B_D(\rho(t)) &= B_{D0} + B_{D1}\rho(t) + B_{D2}\rho^2(t) + B_{D3}\rho^3(t) + B_{D4}\rho^4(t) + B_{D5}\rho^5(t)
\end{aligned} \tag{3.6}
$$

where $\rho \in [0, 1]$ denotes the damage parameter. The above model sufficiently and accurately covers all cases from $\rho = 0$ (no-damage case) to $\rho = 1(100\%$ wing damage).

Control Effectors Damage Faults

These faults are caused by damage in the control effectors of aircraft such as aileron, rudder and elevator surfaces. These faults cause the changes in the gain effectiveness matrix and generate state-dependent disturbance [6]. These faults can be modeled using the diagonal control effectors damage matrix D whose elements d_i are equal to one in the no-fault case while in the case of control effectors damage assumes values over an interval $[\varepsilon, 1]$, where $\varepsilon << 1$,and each value of d_i is proportional to the percentage of the loss of surface. The resulting model is of the form:

$$
\dot{x} = f(x) + g(x)Du + \xi(x) \tag{3.7}
$$

where $D = \left[\begin{array}{cccc} d_1 & d_2 & \cdots & d_m \end{array} \right]$, and $\xi(x)$ is a nonlinearity arising due to the asymmetry of damage aircraft.

Transmitter/Receiver Faults

The communication component faults can be categorized into the following cases:

- **Transmitter (TX) fault:** This may take place due to the electronic malfunctioning of the transmitter device, power blackout in the transmitter unit, or the physical destruction of the transmitter antenna, for example, due to an attack.

- **Receiver (RX) fault:** The fault in the receiver may happen due to the similar reasons as mentioned for the transmitter.

3.2 Information Flow Faults

3.2.1 Issues of UAV Communications in Cooperative UAV Control

With the emergence of the new generation of UAVs, one of the crucial factors to ensure the successful collaboration among multiple UAVs in cooperative missions is the development of reliable, high performance wireless communication links between them. The UAVs communicate relevant information and sensory data and can autonomously reconfigure themselves in response to the shared information and their own states to achieve the mission at hand. For example, in the special case of the autonomous formation flight application, in order for the formation controller to maintain the formation geometry, each aircraft requires its position from a GPS-based position sensing system and receives through appropriate communication channels the other aircraft's positions and velocities. The air-to-air communication system enables the sharing of sensor and map information among UAVs, while the air-to-ground communication system provides mission information and payload data to the ground station for the purposes of mission control and data analysis and display respectively [40]. To be more precise, the UAVs critically depend on communication to:

- Coordinate among themselves in a cooperative mission,
- Obtain high-level commands concerning their mission, tasks, and objectives, and
- Report their observations to human experts or to data repositories in command and control centers.

Thus, due to the critical importance of the inter-UAV communication, acquiring a basic knowledge of the existing wireless communication technologies and challenges involved in the application of these technologies to UAV communications is essentially invaluable. This information will also lead to a profound understanding of communication mechanisms among UAVs which will consequently result in a more comprehensible characterization of communication faults or anomalies. Also, the knowledge gained regarding the UAV communications paradigms can extensively be used in the design of an efficient FDIR scheme for the communication faults that arise in UAV teams.

Wireless Communication Networks

Wires, which enabled the Internet revolution, have now, in many cases, simply become a barrier to proliferation [2]. When communication nodes are un-tethered, essentially new opportunities arise. For example, when a large set of embedded devices are wirelessly connected, we will have a multitude of sensors and actuators all around us that can intelligently communicate and cooperate to fulfill a mission. The type of wireless communication networks that are of interest in UAV communication and we will describe here are currently called *ad hoc* networks. Figure 3.4 shows a general *ad hoc* network with its associated nodes. Each node is equipped with a wireless transmitter/receiver capable of transmitting/receiving data packets to/from its neighbors. Packets can be relayed from node to node until they reach their final destination. Nodes may be mobile, changing their locations over time. Also, they may switch themselves off from time to time.

The wireless medium is unreliable since the transmissions are subject to obstacles, reflections, multi-path effects, and fading. Furthermore, transmissions can interfere with each other, thus making packet reception at the destination more unreliable than in wired media. Clearly, the overall objective of a wireless network is the reliable and efficient transfer of packets from their sources to their destinations. To fulfill this objective, nodes of the network must perform the following tasks:

- Choose the power levels at which they broadcast since that influences the range,

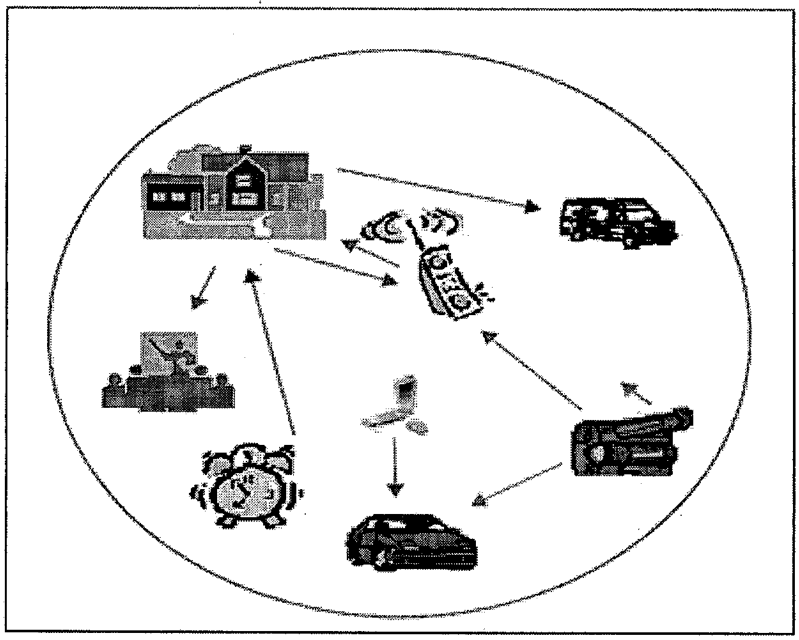


Figure 3.4: An ad hoc wireless network [2]

- Broadcast in a timely fashion to avoid interferences as much as possible, and
- Cooperate in relaying each other's packets.

For the purpose of performance evaluation of a wireless network, two important performance measures are considered as follows:

(i) *Throughput,* which is defined as the rate at which the bits are transferred from the source node to their final destination. It is measured in bits per second.

(ii) *Delay,* which is defined as the time difference from when a bit is transmitted at its source node and eventually received correctly at its destination node.

For real-time interactive services like cooperative UAV teams, timeliness of receipt of packets is extremely crucial. Hence, delay is a highly important performance measure in UAV communication. On the other hand, throughput is of less significance due to the rather small size of the information and control data that need to be transferred among UAVs. There are some other performance indices derived from the above mentioned performance measures such as the mean delay of packets, the standard deviation of the delay, or even the 99 percentile of the delay. All the above performance indices are revealed under the common term of *'Quality of Service (QoS)'*.

The following four problems arise and should be addressed in the design of any wireless network [2]:

- **The Power Control Problem:** A fundamental issue faced by a node in a wireless network on every transmission is how to choose the power level at which it should transmit. As a solution, this problem can be considered as a feedback-based set-point regulation problem. More specifically, the receiver can provide a feedback signal to the transmitter, allowing it to regulate its transmitted power level so that net received signal-to-interference-plus-noise ratio at the receiver is at a desired level.

- **The Medium Access Control Problem:** The issue is that how nodes in a wireless network schedule their transmissions in order to have their packets received intelligibly by their intended receivers; i.e. without any destructive interference between packets due to packet collisions. One scheme that has been proposed towards the solution to this problem is embodied in the IEEE 802.11 standard currently used in Wireless Local Area Networks (WLAN) [29]. This protocol employs reservation packets to reserve the channel locally in space for data packets.

- **The Routing Problem:** Since in an ad hoc network, the address of a node is just simply a name, it does not provide any information regarding the node location or how to reach it. This gives rise to the problem of how to determine the route to be followed by packets from their sources to their given destinations.

- **The Traffic Control Problem:** Given the inherent shared nature of the wireless medium, the question is that what is the ultimate traffic-carrying capacity of wireless networks? The answer will provide a design objective to be satisfied. The performance measure that will be analyzed for this respect is the *transport capacity* of the network that is defined as the aggregate *bit-meters/second* (bit-m/s) that the network as a whole can transport.

Currently, the Wireless Local Area Network (WLAN) with the standard off-the-shelf protocol of IEEE 802.11 is in extensive use in developed societies. Also Wireless Metropolitan Area Network (WMAN) with the newly developed IEEE 802.16 standard protocol is getting into the US market. The main difference between these two networks is the maximum allowable distance between two nodes in the network. While in WLAN the maximum distance is almost 100 meters in an obstacle free environment, in the case of WMAN this distance is extended to the order of kilometers. In terms of the throughput, some WLAN protocols like IEEE 802.11g can provide up to 54Mbps rates.

Communication Networks in Cooperative UAVs

In a cooperative UAV control, each UAV participates in two networks [40]:

- An external wireless network that enables the coordination and communication of the UAV with other entities, such as other UAVs or ground stations, and

- An internal network that connects on-board sensors, actuators and other devices.

For the internal network, any of the existing commercial industrial (wired) network types such as Ethernet, ControlNet, or DeviceNet can be used. Especially DeviceNet with the CAN (Control Area Network) protocol is of high interest. CAN [41] is a serial communication protocol developed mainly for applications in the automotive industry but is also capable of offering good performance in other time-critical applications like UAVs. However, the internal network of a UAV is not the research issue of interest in our project. Basically, as was mentioned earlier, we are interested in inter-UAV communications (external network) fault detection and recovery.

UAV-to-UAV wireless communication, however, is a more recent problem and is currently implemented with proprietary, application-specific, and inflexible protocols that use point-to-point, fixed-bandwidth channels. The challenge is to enable UAV communication that supports scalable, evolvable, and intelligent protocols. To the best of our knowledge, to date there is no field-tested COTS (Commercial Off-the-Shelf) solution for this application. The difficulties include finding antennas with suitable omni-directional patterns and low power requirements. Inter-UAV communication may also be required in out of line-of-sight conditions due to distance or obstacles. Furthermore, the possibility of relatively large distances among UAVs and more importantly the very high relative speed of the UAVs (the mobile nodes of the communication network) make the implementation of a reliable wireless communication scheme more challenging. Some studies [42] have recently been done on performance evaluation of the application of WLAN (IEEE 802.11a) and WMAN (IEEE 802.16) networks, using OFDM (Orthogonal Frequency Division Multiplexing) based multi-carrier transmission technology [43], to UAV wireless communication. However, as stated therein, the wireless communication channel in UAV communications is largely different from other "traditional" wireless channels in the fact that in the later ones, one end (the base station) is fixed and the relative speed of the other end (mobile terminal) to base station is hardly over 80 miles per hour. By Contrast, in UAV communications, the relative speed of the both ends of the communication system could be as high as 500 miles per hour. Such high mobility leads to much shorter coherence time and a larger Doppler spread [43] in the multi-path fading channel. Thus, the Inter-Carrier-Interference (ICI) is much larger than those of WLAN and WMAN scenarios. Nonetheless, based on the simulation results reported in [42], the Bit Error Rate (BER) performance of the OFDM based UAV communications, i.e. at high relative speeds of UAVs, is still comparable to the traditional OFDM based WLAN networks. Thus, the authors [42] conclude that with a proper selection of its parameters, wireless OFDM based technology would be a feasible choice for next generation UAV communications.

UAV Cooperative Control with Communication Constraints

Even in the presence of a highly reliable and efficient communication network among UAVs, there are always some information flow constraints such as bandwidth limitations, delays, and range and network topology limitations of the inter-UAV communication. These constraints and their effects on the cooperative UAV control have been studied by researchers in [44], [45], [46].

In [44], the authors have developed algorithms for team-optimal and individually-optimal/team-suboptimal solutions for multiple UAV cooperative search problem subject to the constraints of collision-avoidance and that the UAVs should stay within communication range of one another. More specifically, the control objective for the team is to maximize the regions of opportunity visited by the team, while minimizing the regions of hazard, subject to two path constraints: (1) that the communication network remains connected at all times, and (2) that there are no collisions between UAVs.

In [46], the author has considered the UAV cooperative control problem with information flow constraints. Three different form of communication constraints are modeled that represent communication of information between UAVs as a sequence of impulses, a band-width limited signal, and a range-limited signal. Then the UAV cooperative control problem with the above information flow constraints is formulated as a series of generalized optimal control problems by considering the inter-UAV communication as another control input. The three proposed models for the information flow constraints are as follows:

- **Communication with constraints among UAVs as a sequence of impulses**

Here the information flow constraints among UAVs is modeled as a sequence of impulses

$$v(t) = \sum_i c_i \delta(t - t_i),\ t_{i+1} - t_i \geq T \tag{3.8}$$

where the time interval between two impulses is greater than some constant T. Then the task planning tasks of UAV cooperative control is with the above communication constraint is formulated to be a generalized optimal control tasks with the constraint in (3.8), that is,

$$\min_{u,v} J = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x, u, t)dt \tag{3.9}$$

Subject to the constraints,

$$\dot{x}(t) = f(x, u, t) + h(x, u, t)v \tag{3.10}$$

$$v(t) = \sum_i c_i \delta(t - t_i) \tag{3.11}$$

- **Communication with constraints among UAVs as a bandwidth-limited signal**

Here the information flow and communication constraints between UAVs is modeled as a function

$$v(t) = \sum_i g(t)(1(t - t_{2i}) - 1(t - t_{2i+1})),\ t_{j+1} - t_j \geq T \tag{3.12}$$

where $g(t)$ is a piecewise continuous function, and again the time interval between information flow is greater than some constant T. Again the UAV cooperative control is formulated as in part (a) with the constraint (3.11) replaced by the new constraint (3.12).

- **Communication with constraints among UAVs as a range-limited signal**

Here, the information flow constraints over a wide geographical area with geographical constraints is modeled as a range-limited function

$$v(t) = \begin{cases} g(t),\ C(x) > 0 \\ 0\ ,\ C(x) \leq 0 \end{cases} \tag{3.13}$$

where again $g(t)$ is a piecewise continuous function, and $v(t)$ is limited information from outside the UAV. Again the UAV cooperative control problem is formulated as before but with the new constraint in (3.13).

The necessary conditions for the existence of the optimal control actions for all the three above constrained optimal UAV cooperative control problems are also derived by the author [7].

In [46], the authors have considered the problem of cooperative search and task response in a team of UAVs with limited communications. The UAVs are engaged in a mission to search and verifiably destroy targets in an uncertain environment while they cooperate in two ways: 1) by sharing information among the team; and 2) by coordinating their tasks. The decentralized system described by the authors requires the specification of three components:

(i) **The information-sharing policy (ISP):** Which should answer the fundamental question of: Who communicates what to whom, when and how?

(ii) **The information fusion policy (IFP):** Which answers the question of: How is received information combined with the existing subjective information base (SIB)?

(iii) **The decision-making algorithm:** Which answers: How does each UAV use its SIB to make its decisions?

The main contribution the above work in [46], is the definition and devise of the above three components (i.e. ISP, IFP, and decision-making algorithm) with the objective of bringing the UAV team's performance as close as possible to that achieved by the centralized algorithm with noise-free, instantaneous communication (i.e. unconstrained communication).

Graph Theory Representation of the Communication Topology

In the work presented in [42], [45], the authors propose a graph theory representation for information exchange and communication among UAVs. Although the work is done under the special case of decentralized formation flight control and management but we believe that the same idea could be extended to the more general case of UAV cooperative teams. In this paradigm, the information exchange between UAVs can be point-to-point or broadcast. By contrast to the former case, in the latter one, every UAV receives and sends data to all the others. However, in both cases, the UAVs can be though of as the *nodes* of a graph while the physical communication channels create the *arcs* in the graph. The arcs are assumed to be oriented since, in the most general case, channels are not bidirectional. The communication graph should be redundant from the viewpoint of the information flow capability. This redundancy makes reconfiguration feasible in the event of communication faults. Some of the possible reconfiguration techniques are described in the later sections.

Optimal Communication Channels among UAVs

Having the capability of using a channel does not mean that it must be used at all times. In other words, among the many different configurations that can be found in a UAV communication graph, only a few of them may provide good performance. Thus, even without any communication faults, there should be some algorithm for the specification of the optimal selection of the available communication channels to be used for propagating information among the UAVs in the formation. The authors in [18] have proposed an approach for communication channels optimization via graph programming techniques.

To implement the optimization technique, a weight must be assigned to each arc in the graph. Then the optimization algorithm will minimize the total cost of the information paths throughout the formation using the arcs' weights to evaluate the cost of a connection. The arcs' weights can be chosen depending on various criteria such as:

- Capability of the formation control system to maintain constant inter-UAV distances. A measure of control effort or closed-loop distance-keeping performance can be used to set the link weight.
- Formation safety: using distance references with adjacent UAV might limit the risk of UAV conflicts compared to using a common reference for all UAVs.
- If a non-radio-based or a non-omni directional communication channel is used, the geometry of the formation might influence the possibility of exchanging data between two UAVs that are not closely spaced or hidden by the others.

Authors in [47] have proposed the deterministic Dijkstra algorithm for the solution of the communication channels optimization problem.

3.2.2 FDIR of Information Flow Faults in Cooperative UAVs

Although the design of a reliable, high performance wireless communication scheme for cooperative UAV control is still a quite challenging problem, however, from the control engineering point of view, the research issue of interest in this project is the design of a fault tolerant control scheme using the best available wireless communication tools. This fault tolerant scheme should be capable of (i) detecting any communication fault autonomously and (ii) autonomously reconfiguring the mission controller or the communication channel among the UAVs, using any of the existing hardware or software redundancies, to fulfill the mission objectives. However, there are still only a few works reported in the literature regarding the design of an efficient FDIR scheme for the communication faults of a cooperative UAV control (i.e. fault-tolerant UAV control) problem. The major works that have been reported, and will be described in more detail in section 3.2.2, are in the specific area of flight formation. These include the works done in [47], [18] in communication channel reconfiguration in the presence of some classes of communication faults and the work done in [18] which addresses the communication blackout detection and isolation in the formation flight control problem. It should be emphasized that to the best of our knowledge there is no work reported in the literature to date, that considered the FDIR issue in UAV cooperative control problem in the presence of both low-level and communication faults.

Characterization of Information Flow Faults

In the most general terms, a *communication fault* can be defined as the temporary or permanent loss of information exchange capability in one or more UAVs in a cooperative UAV control problem. The information exchange (or communication) capability is provided by transmitter (TX) and receiver (RX) devices on each UAV. It is also clear that the physical loss of a UAV itself is a communication fault since both the receiving and sending information capability are lost. Based on the above definition, we can broadly classify the communication faults into the two following categories:

- **Temporary faults:** The communication capability may be lost for only a limited period of time. The time duration of temporary fault, however, is an important factor in the cooperative control problem. The temporary faults may happen due to several reasons such as the presence of obstacles between UAVs, the very large distances (larger than the specified communication range) that may temporarily exist among UAVs, the existence of highly noisy environments or the radar jamming (conducted by the enemy) in only some sub-areas of the terrain that the UAV teams operate in, or un-successful transmission or receiving of a data packet for limited sampling times.

- **Permanent faults:** The communication capability of one or more nodes in the UAV team is lost for the rest of the mission. In the presence of such kind of faults, especially in the case of UAV flight formation, some UAV reconfiguration algorithm like the one presented in [36] should be initiated in order to maintain the team operative. The permanent faults may occur due to several factors such as the permanent loss of the receiver, the permanent loss of the transmitter, the loss of the UAV, or the sudden initiation of radar jamming (conducted by enemy) in *almost all* the terrain that the UAVs conduct their mission.

From another viewpoint, the communication faults can be originated due to the following sources of faults:

- **Transmitter (TX) fault:** This may take place due to the electronic malfunctioning of the transmitter device, power blackout in the transmitter unit, or the physical destruction of the transmitter antenna, for example, due to an attack.

- **Receiver (RX) fault:** The fault in the receiver may happen due to the similar reasons as mentioned for the transmitter.

- **Partial/Full data packet loss:** This type of fault represents the missing of the full data packet for some sampling times in the receiver end although it has been sent correctly by the transmitter, or the loss or change of only some bits/symbols in the data packet arriving at the receiver end. This type of fault may happen due to some stochastic sources like the high levels of environmental noise or deterministic reasons like radar jamming conducted by the enemy. In the case of partial loss of data, the significance of the bits/symbols lost plays a crucial role in the UAV cooperative control problem. For example, in a formation flight control problem, the loss of bits or symbols representing some other UAVs' positions data can cause significant difficulties for the formation controller while the loss of some other bits may not be that significant.

- **UAV loss:** This type of fault essentially happens due to the physical destruction of the whole UAV. Thus, one node of the inter-UAV communication network would be lost forever.

Physical Implication of Faults In section 3.2.2, we tried to justify the physical implications and sources of different types of communication faults in a rather comprehensive way. As stated therein, the radar jamming is considered as the source of many different communication fault scenarios. So it might be helpful to gain a basic understanding of the radar jamming mechanisms. In general, the techniques that limit the effectiveness of an opponent's communications and/or detection equipment are called 'jamming' [43]. The jamming techniques can be broadly classified into the following three classes of applications:

- Radio jamming
- *Radar jamming*
- Cell phone jamming

Of particular interest to UAV team communication is the radar jamming technique(s). By definition [43], *Radar jamming* is the intentional emission of radio frequency signals to interfere with the operation of a radar by saturating its receiver with false information. There are basically two types of radar jamming, namely, mechanical and electronic jamming.

Mechanical jamming is caused by devices which reflect or re-reflect radar energy back to the radar to produce false target returns on the operator's scope. Mechanical jamming devices include chaff, corner reflectors, and decoys [43].

Electronic jamming is a form of electronic attack where jammers radiate interfering signals toward the enemy's radar blocking the receiver with highly concentrated energy signals. The four existing types of electronic jamming are *spot, sweep, barrage,* and *repeater/spoofer* jamming [43].

Information Flow Fault Recovery

After a fault, one or more connections could be lost and a new channel configuration must be found or otherwise the cooperative control tasks should be reconfigured to recover the mission. The algorithm for the re-optimization of the inter-UAV connection and the control reconfiguration algorithm are triggered by the fault detection, and they are essentially decentralized for faster reconfiguration time. Also, only the information that is strictly needed for the reconfiguration process must be exchanged on the operating data channels. The reconfiguration process must arrive to the same result on all UAVs; that is, the local copy of the graph describing the information communications must be identical in all UAVs at all times.

There are some cases (like the UAV loss fault), however, that after a reconfiguration of the inter-UAV connections, a geometrical reconfiguration of the UAVs is needed as well. The authors in [47] have described a set of heuristic rules, implemented through several schemes, called reconfiguration maps (RMs) for the purpose of geometrical reconfiguration of the formation. Although their work has been done for a special case of leader-follower formation in the UAV formation control problem, the same idea can be extended to a cooperative UAV control problem.

The Hardware Redundancy Considerations In the event of communication fault some hardware redundancies are usually predicted in order to provide information regarding at least the other UAVs' positions which is variable of critical importance in cooperative UAV control. These redundancies are implemented through on-board AFF (Autonomous Formation Flyer) type sensors that measure inter-UAV distances. In other words, in the presence of a transmission fault the follower estimates the state of the leader using its current estimate and the measurement of their relative position provided by onboard sensors. In the case of UAV formation flight the measurement of the relative position can also be provided by the use of sensors such as camera, infra-red, or laser mounted on the follower vehicle and a target such as pattern, light, or reflector on the lead UAV. Although this requires more additional hardware but it is simple, robust, and immune to jamming. Also differential carrier phase GPS can be used to get relative position with more precision. For example, in the GTMax research UAV system [39] developed at the Georgia Institute of Technology to support the Software Enabled Control (SEC) program launched by DARPA and AFRL, the basic system hardware includes a general-purpose computer, Differential GPS (D-GPS), an inertial measurement unit, an ultrasonic altimeter, a three-axis magnetometer, and two wireless data links. Other flight configurations used to date have also utilized cameras, a radar altimeter, and video capture/compression hardware. The basic ground equipment includes the data links and a GPS reference station.

Software Redundancy & Communication Channel Reconfiguration Although the hardware redundant sensors can be used in the case of communication fault, but in many situations they are not considered as the first choice of communication fault recovery. In other words, it is more desirable to exploit the software redundancies as the first solution. For example, in order to decrease the probability of partial data packet loss, a data packet can be sent more than once using the possible extra bandwidth of the communication channel. Obviously, the time between two transmissions of a single data pack should be smaller than the sampling time of the sensory data of the UAVs. Authors in [18] have also proposed the issue of redundant channels as a recovery technique to the occurrence of communication faults of TX fault, RX fault, and UAV loss in the autonomous flight formation problem. They use a graph theory approach (modified Dijkstra algorithm) to optimally reconfigure the communication channel after a fault has occurred. Using this algorithm the optimal redundant communication channel among UAVs is found. However, they still use a separate Broadcast Channel (BC) to keep all the UAVs informed of what is happening on a specific node of the UAVs communication network [47]. The information carried on the BC is in fact vital to the coordination of the formation whenever required (for example, in case of a UAV loss). Thus, the use of BC communication is quite more extensive during a post-fault recovery procedure. The authors in [48], have also proposed the inter-UAV communication Fault Detection and Isolation (FDI) scheme based on the Interacting Multiple Model (IMM) approach. However, the class of communication faults that they consider is limited to the general case of communication blackout. Furthermore, they don't propose any fault recovery algorithm for post fault operation of UAVs.

Chapter 4

Model-Based Fault Detection, Isolation and Recovery

4.1 Nonlinear Geometric Fault Detection and Isolation Approach

In the following, the method presented in [49] is briefly reviewed, which gives us the necessary and sufficient conditions for solving the problem of detecting and isolating faults for nonlinear systems. The reader may refer to [49] for all the details and further information. The mathematical concepts and definitions may be found in [50]. The essential parts of the work in [49] are reviewed here. It is assumed that the nonlinear system can be described by the following model:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u + l(x)m_1 + p(x)w \\ y &= h(x)\end{aligned} \tag{4.1}$$

with state x defined in a neighborhood X of the origin in R^n, inputs $u \in R^m$, $m_1 \in R$, $w \in R^d$ and output $y \in R^q$, in which $f(x)$, the m columns $g_1(x), ..., g_m(x)$ of $g(x)$, $l(x)$, and d columns $p_1(x), ..., p_d(x)$ of $p(x)$ are smooth vector fields, $h(x)$ is a smooth mapping and $f(0) = 0$, $h(0) = 0$. The notations u, m_1, and w of the input of (4.1) denote the input channel for the control purposes, the fault signal, and the disturbance signal, respectively.

By ignoring the faults and disturbances momentarily, then (4.1) becomes

$$\begin{aligned}\dot{x} &= g_0(x) + \sum_{i=1}^{m} g_i(x)u_i \\ y &= h(x)\end{aligned} \tag{4.2}$$

where $g_0(x) = f(x)$.

Definition 4.1.1. Let U be an open set in R^n. Then the set of *smooth functions with compact support* (in U) is the set of functions $f : R^n \to C$ which are smooth (i.e., $\partial^\alpha f : R^n \to C$ is a continuous function for all multi-indices α) and $\text{supp}(f)$ is compact and contained in U [51].

Definition 4.1.2. Suppose U is an open set in R^n, and suppose $D(U)$ is the topological vector space of smooth functions with compact support. A *distribution* is a linear continuous functional on $D(U)$, i.e., a linear continuous mapping $D(U) \to C$ [51].

Definition 4.1.3. Let $T : V \to W$ be a linear transformation. The set of all vectors in V that T maps to 0 is called the *kernel* (or *nullspace*) of T, and is denoted $KerT$. When the transformations are given by means of matrices, the *kernel* of the matrix A is

$$KerA = \{x \in V \,|Ax = 0\}$$

Definition 4.1.4. Given a distribution Δ, the distribution that is the smallest involutive distribution containing Δ is called the *involutive closure* of Δ [50].

Corresponding to (4.2), a distribution Δ is called conditioned invariant if it conforms

$$[g_i, \Delta \cap Ker\,\{dh\}] \subset \Delta, \qquad i = 0, \ldots, m \tag{4.3}$$

Further, we let $P = span\,\{p_1(x), ...p_d(x)\}$, and propose a nondecreasing sequence of distributions as following:

$$\begin{aligned} S_0 &= \bar{P} \\ S_{k+1} &= \bar{S}_k + \sum_{i=0}^{m} [g_i, \bar{S}_k \cap Ker\,\{dh\}] \end{aligned} \tag{4.4}$$

where $\bar{S}$ denotes the involutive closure of S. Suppose there is an integer k^* such that

$$S_{k^*+1} = \bar{S}_k \tag{4.5}$$

and let $\sum_*^P = \bar{S}_{k^*}$. Thus, $\sum_*^P$ is involutive, contains P and is conditioned invariant.

Let Θ be a fixed codistribution and define the following nondecreasing sequence of codistributions

$$\begin{aligned} Q_0 &= \Theta \cap span\,\{dh\} \\ Q_{k+1} &= \Theta \cap \left(\sum_{i=0}^{m} L_{g_i} Q_k + span\,\{dh\} \right) \end{aligned} \tag{4.6}$$

Suppose all the codistributions of this sequence are nonsingular, so that there is an integer $k^* \leq n-1$ such that $Q_k = Q_{k*}$ for $k > k^*$, and set $\Omega^* = Q_{k^*}$. Then, it is convenient to use the notation

$$\Omega^* = o.c.a.\,(\Theta)$$

where "*o.c.a.*" stands for "observability codistribution algorithm".

Based on the properties (as discussed in details in [49]) of the two sequences (4.4) and (4.6), if (4.6) is initialized at $\left(\sum_*^P\right)^\perp$, then $o.c.a. \left(\left(\sum_*^P\right)^\perp\right)$ is the largest observability codistribution which is locally spanned by the exact differentials and contained in $P^\perp$.

According to [49], the necessary condition for solving the nonlinear fundamental problem of residual generation is:

$$(span\,\{l\})^{\perp} + o.c.a.\left(\left(\sum\nolimits_{*}^{P}\right)^{\perp}\right) = T^{*}X \tag{4.7}$$

Furthermore, if the condition (4.7) holds, then we could possibly perform coordinate transformations in the states leading to the new output that characterizes an observable subsystem upon which it is possible to design a residual generâtor unaffected by all faults but one.

Definition 4.1.5. Suppose there is an open subset U of R^n. The *differential* or *gradient* of a real-valued function $\lambda(x)$ on U, denoted $d\lambda$, is the $1 \times n$ row vector whose i-th element is the partial derivative of λ with respect to x_i. Its value at a point x is thus

$$d\lambda(x) = \begin{pmatrix} \dfrac{\partial\lambda}{\partial x_1} & \dfrac{\partial\lambda}{\partial x_2} & \cdots & \dfrac{\partial\lambda}{\partial x_n} \end{pmatrix}$$

Definition 4.1.6. A function $f : X \to Y$ is called *surjection* if, for every $y \in Y$, there is an $x \in X$ such that $f(x) = y$.

Consider the nonlinear system dynamics that is governed by (4.2). Let Ω be an observability codistribution, and n_1 be the dimension of Ω. When Ω is calculated according to the sequences (4.4) and (4.6), (4.6) is initialized at $\left(\sum_{*}^{P}\right)^{\perp}$, and Ω can be spanned by the exact differentials.

Suppose that $span\,\{dh\}$ is nonsingular. Let $p - n_2$ denote the dimension of $\Omega \cap span\,\{dh\}$, and suppose there exists a surjection $\Psi_1 : R^p \to R^{p-n_2}$ such that

$$\Omega \cap span\,\{dh\} = span\,\{d\,(\Psi_1 \circ h)\} \tag{4.8}$$

Fix $x^{\circ} \in X$ and let $y^{\circ} = h\,(x^{\circ})$. Then, there exists a selection matrix H_2 such that

$$\Psi\,(y) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \Psi_1\,(y) \\ H_2 y \end{pmatrix} \tag{4.9}$$

Choose a neighborhood U° of x° and a function Φ_1: $U^{\circ} \to R^{n_1}$ such that

$$\Omega = span\,\{d\Phi_1\} \tag{4.10}$$

at any point of U°. Then, there exists a function $\Phi_3 : U^{\circ} \to R^{n-n_1-n_2}$ such that

$$\Phi\,(x) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \Phi_1\,(x) \\ H_2 h\,(x) \\ \Phi_3\,(x) \end{pmatrix} \tag{4.11}$$

is a local diffeomorphism at x° in X. By applying this diffeomorphism to system (4.1), then system (4.1) becomes equivalently:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) + g_1(x_1, x_2)u + l_1(x_1, x_2)m_1, x_1 \in R^{n_1} \\ \dot{x}_2 &= f_2(x_1, x_2, x_3) + g_2(x_1, x_2, x_3)u + p_1(x_1, x_2, x_3)w, x_2 \in R^{n_2} \\ \dot{x}_3 &= f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3)u + p_2(x_1, x_2, x_3)w, x_3 \in R^{n-n_1-n_2} \\ y_1 &= h_1(x_1), y_1 \in R^{p-n_2} \\ y_2 &= x_2, y_2 \in R^{n_2} \end{aligned} \tag{4.12}$$

Following the state space coordinate transformation, the disturbances w can only affect x_1-subsystem through x_2. However, since we observe x_2 through y_2, one may view x_2 as an input to the x_1-subsystem. The problem of detecting and isolating faults in the original nonlinear system (4.1) reduces to that of designing a residual generator for the observable subsystem [49]. However, either designing an estimator for the states of the system or its parameters require that the states of the system be uniformly bounded. Thus, if the open-loop system is not guaranteed to be state bounded, one has to design a controller to make the system stable and state bounded. More importantly, when actuator faults occur, it is clear that the existing controllers will not work and the states will diverge most likely. Therefore, it is imperative that as soon as one detects the faults, one initiates the trigger for the redundant actuators for compensating the presence of faults. The difficulty that one is faced with is that one cannot control the system from the time the faults occur to the time the faults are detected. Therefore, there is a necessary condition for designing a residual generator for the observable subsystem. Specifically, the system should contain no finite escape time.

Based on the above review of the fault diagnosis of nonlinear systems, the main contribution of this thesis is to design a desirable residual generator for the observable subsystem. Our approach is based on parameter estimation. The actuator faults are viewed and considered as changes of the parameters. A nonlinear estimator is designed to identify these parameters and the estimators are treated as the residual generators. The proposed new residual generators are applied to two different dynamical models namely, satellite orbital model and satellite attitude control model.

4.1.1 Isolation of Concurrent Faults

For the nonlinear system (4.1), there is only one possible fault m_1. In this section, we will discuss the case in which there are possibilities of multiple concurrent faults. The key idea of isolating concurrent faults is to design a residual generator for each possible fault, where this residual generator is unaffected by all the faults except one fault. Consider the system

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x)u_i + \sum_{i=1}^{s} l_i(x)m_i + p(x)w \\ y &= h(x) \end{aligned} \tag{4.13}$$

where there are now s possible faults. For i=1,..., s, set

$$\begin{aligned} \tilde{p}_i &= (l_1 \quad \cdots \quad l_{i-1} \quad l_{i+1} \quad \cdots \quad l_s \quad p) \\ \tilde{w}_i &= col\,(m_1, \cdots, m_{i-1}, m_{i+1}, \cdots, m_s, w) \end{aligned}$$

Then, system (4.13) can be rewritten as

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x)u_i + l_i(x)m_i + \tilde{p}_i(x)\tilde{w}_i \\ y &= h(x) \end{aligned} \tag{4.14}$$

Let $\tilde{P}_i = span\,\{\tilde{p}_i\}$, so that we can use nondecreasing sequence (4.4) and (4.6), where (4.4) is initialized at $\tilde{P}_i$ and (4.6) is initialized at $\left(\sum_*^{\tilde{P}_i}\right)^\perp$, to find the largest observability codistribution which is locally spanned by the exact differentials and contained in $\tilde{P}_i^\perp$. Further, if this observability codistribution satisfies (4.7), we may transform the coordinates to have an observable subsystem upon which it is possible to design a residual generator affected only by m_i. If one can design such a residual generator for each m_i (i=1,..., s), one in effect can isolate these concurrent faults.

4.1.2 Step by Step details of the Nonlinear FDI approach

In this section, design steps of nonlinear geometric FDI algorithms are summarized. Consider nonlinear system 4.1 and a problem of generating a residual signal that is affected by fault m_1 and is decoupled from w. Design step for FDI algorithm for this problem may be summarized as:

(i) For $P = span\,\{p_1(x), ...p_d(x)\}$, find $\sum_*^P$ using a nondecreasing sequence of distribution in (4.4).

(ii) Find the observability codistribution $o.c.a((\sum_*^P)^\perp)$ using (4.6).

(iii) Check the necessary condition (4.7), if it does not hold, the problem does not have a solution.

(iv) If (4.7) holds, then one should find a function Φ_1 that satisfies (4.10). This function can be found solving the corresponding partial differential equation. For example let $\Omega = o.c.a((\sum_*^P)^\perp) = P^\perp$ with dimension n_1 (this will happen when there is full state measurement), then one should solve following PDEs

$$\begin{aligned} \partial\lambda(x).p_1(x) &= 0 \\ \partial\lambda(x).p_2(x) &= 0 \\ &\vdots \\ \partial\lambda(x).p_d(x) &= 0 \end{aligned} \tag{4.15}$$

Since Ω is involutive, there should be n_1 solutions $\lambda_1(x), \lambda_2(x), ..., \lambda_{n_1}(x)$ for above PDE. Then

$$\Phi_1(x) = \begin{pmatrix} \lambda_1(x) \\ \lambda_2(x) \\ \vdots \\ \lambda_{n_1}(x) \end{pmatrix} \tag{4.16}$$

(v) Find a surjection Ψ_1 that satisfy (4.8) and selection matrix H_2 such that Ψ is output diffeomorphism.

(vi) Find the state diffeomorphism Φ from (4.11) and the new state equation of the system (4.1) as (4.12).

(vii) Design an observer for the x_1 subsystem of (4.12) by considering $x_2 = y_2$ as an independent input.

4.2 Fixed-Wing UAV FDI

4.2.1 Fixed Wing UAV Models

References [3, 52] discuss the basic concepts of the fixed-wing aircraft dynamics. In this subsection, the axes of aircraft motions, main control surfaces, and fixed-wing models will be presented.

Dynamic Motions [3]

The unsteady motions of an airplane can frequently be separated for convenience into two parts. One of these consists of the longitudinal or symmetric motions; that is, those in which the wings remain level, and in which the center of gravity moves in a vertical plane. The other consists of the lateral or asymmetric motions; that is, rolling, yawing, and sideslipping, while the angle of attack, the speed, and the angle of elevation of x axis remains constant.

Main Control Surfaces

The main control surfaces are attached to the airframe on hinges so they may move and deflect the air stream passing over them. This redirection of the air stream generates an unbalanced force to rotate the plane about the associated axis.

- Ailerons - Ailerons are mounted on the back edge of each wing near the wingtips, and move in opposite directions. When the pilot moves the stick left, or turns the wheel counter-clockwise, the left aileron goes up and the right aileron goes down. A raised aileron reduces lift on that wing and a lowered one increases lift, so moving the stick left causes the left wing to drop and the right wing to rise. This causes the plane to bank left and begins to turn to the left. Centering the stick returns the ailerons to neutral maintaining the bank angle. The plane will continue to turn until opposite aileron motion returns the bank angle to zero to fly straight.

- Elevators - An elevator is mounted on the back edge of the horizontal stabilizer on each side of the fin in the tail. They move up and down together. When the pilot pulls the stick backward, the elevators go up. Pushing the stick forward causes the elevators to go down. Raised elevators push down on the tail and cause the nose to pitch up. This makes the wings fly at a higher angle of attack which generates more lift and more drag. Centering the stick returns the elevators to neutral and stops the change of pitch.

- Rudder - The rudder is mounted on the back edge of the fin in the tail. When the pilot pushes the left pedal, the rudder deflects left. Pushing the right pedal causes the rudder to deflect right. Deflecting the rudder right pushes the tail left and causes the nose to yaw right. Centering the rudder pedals returns the rudder to neutral and stops the yaw.

Other Controls

- Flaps - Flaps are mounted on the back edge of each wing near the wing roots. They are deflected down to increase the effective curvature of the wing and produce additional lift, and also reduce the stalling speed of the wing. They are used during low speed, high angle of attack flight like descent for landing.

- Spoilers - On very high lift/low drag aircraft like sailplanes, spoilers are used to disrupt airflow over the wing and greatly reduce the amount of lift. This allows a glider pilot to lose altitude without gaining excessive airspeed. Spoilers are sometimes called "lift dumpers".

- Elevon - An elevon is an aircraft control surface that combines the functionality of the elevator (used for pitch control) and the aileron (used for roll control), hence its name. It is frequently used in tailless aircraft such as flying wings. There will be one or more elevons on each side of the aircraft at the trailing edge of the wing. When moved in the same direction (up or down) they will cause a pitching force to be applied to the airframe. When moved differentially, (one up, one down) they will cause a rolling force to be applied. These forces may be applied simultaneously by appropriate positioning of the elevons.

References [3, 52] investigate the fixed-wing aircraft dynamic models. Here, the results of their investigations will be presented. For simplicity, the treatment will be limited to the flat-Earth equations of motion. First, the vector equations will be expanded with the translational-velocity state equation expressed in terms of velocity components in the aircraft body-fixed system. The body-axes equations are the best choice for general flight simulation. On the other hand, for the purposes of linearizing the equations of motion and studying the dynamic behavior, it is better to have the velocity equation in terms of stability or wind-axes variables: airspeed and aerodynamic angles.

Body-axes equations

Force equations

$$\begin{aligned} X - mg\sin\theta &= m(\dot{u} + qw - rv) \\ Y + mg\cos\theta\sin\phi &= m(\dot{v} + ru - pw) \\ Z + mg\cos\theta\cos\phi &= m(\dot{w} + pv - qu) \end{aligned} \tag{4.17}$$

Kinematics equations

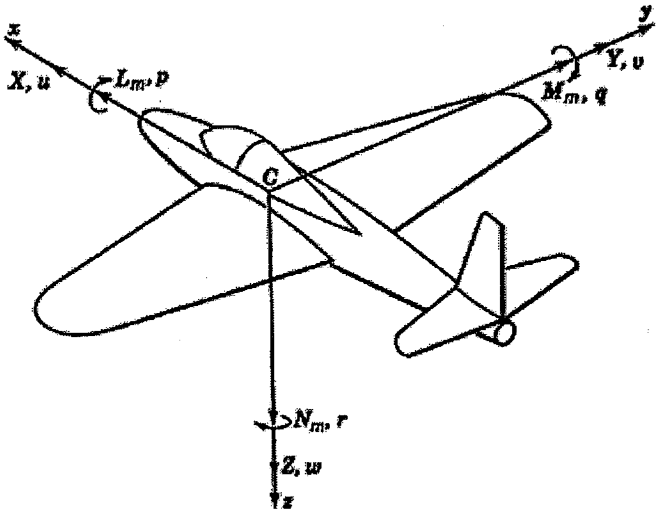


Figure 4.1: Notations for body axes [3]

$$
\begin{aligned}
\dot{\phi} &= p + \tan\theta \cdot (q \sin\phi + r\cos\phi) \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\varphi} &= (q\sin\phi + r\cos\phi)/\cos\theta
\end{aligned}
\tag{4.18}
$$

Moment equations

$$
\begin{aligned}
L_m &= I_x\dot{p} + I_{zx}\dot{r} - qr(I_z - I_y) + I_{zx}pq \\
M_m &= I_y\dot{q} + rp(I_x - I_z) - I_{zx}(p^2 - r^2) \\
N_m &= I_z\dot{r} + I_{zx}\dot{p} + pq(I_y - I_x) - I_{zx}qr
\end{aligned}
\tag{4.19}
$$

Navigation equations

$$
\begin{aligned}
\dot{x} &= \cos\theta \cdot \cos\varphi \cdot u + (\sin\phi \cdot \sin\theta \cdot \cos\varphi - \cos\phi \cdot \sin\varphi) \cdot v \\
&\quad + (\cos\phi \cdot \sin\theta \cdot \cos\varphi + \sin\phi \cdot \sin\varphi) \cdot w \\
\dot{y} &= \cos\theta \cdot \sin\varphi \cdot u + (\sin\phi \cdot \sin\theta \cdot \sin\varphi + \cos\phi \cdot \cos\varphi) \cdot v \\
&\quad + (\cos\phi \cdot \sin\theta \cdot \sin\varphi - \sin\phi \cdot \cos\varphi) \cdot w \\
\dot{z} &= -\sin\theta \cdot u + \sin\phi \cdot \cos\theta \cdot v + \cos\phi \cdot \cos\theta \cdot w
\end{aligned}
\tag{4.20}
$$

where m is the mass of aircraft, x, y, and z are positions of the center of gravity along the x, y, and z-axis directions, u, v, and w are the velocities in the x, y, and z-axis directions, ϕ, θ, and φ are roll, pitch, and yaw angles, p, q, and r are the roll, pitch, and yaw rate, X, Y, and Z are total forces along the x, y, and z-axis directions, and L_m, M_m, and N_m are total rolling, pitching, and yawing moments, as shown in Fig. 5. Also, the inertia matrix of an aircraft is defined as:

$$I_B = \begin{bmatrix} I_x & I_{xy} & I_{xz} \\ I_{yx} & I_y & I_{yz} \\ I_{zx} & I_{zy} & I_z \end{bmatrix}$$

Longitudinal Model in Wind or Stability-Axes Equations

Most aircrafts spend their flying time in a wings-level steady-state flight condition, and due to time limitations, only the longitudinal model [52] in a wings-level and zero sideslip angle conditions will be investigated in this project. Specifically, the equations are given as follows:

$$\begin{aligned} m\dot{V}_T &= F_T\cos(\alpha+\alpha_T) - D - mg\sin(\gamma) \\ m\dot{\gamma}V_T &= F_T\sin(\alpha+\alpha_T) + L - mg\cos(\gamma) \\ \dot{\alpha} &= q - \dot{\gamma} \\ \dot{q} &= M_m/I_y \end{aligned} \tag{4.21}$$

where I_y denotes the y-body axis moment of inertia, m denotes the total mass of the aircraft, α denotes angle of attack, F_T denotes the engine thrust, α_T represents the angle formed by the vector of the engine thrust and the fuselage reference line, γ denotes the flight-path angle, q is the pitch rate, V_T represents the total velocity of the aircraft, M_m denotes pitching moments and L and D represent the lift and drag, respectively.

The lift, the drag and the pitching moments can be calculated according to:

$$\begin{aligned} D &= \bar{q}SC_D \\ L &= \bar{q}SC_L \\ M_m &= \bar{q}S\bar{c}C_m \end{aligned} \tag{4.22}$$

where S denotes the wing platform area, $\bar{c}$ represents the mean aerodynamic chord, C_D, C_L, and C_m are drag, lift and pitching coefficients, respectively, and $\bar{q}$ denotes the dynamic pressure, which is defined by:

$$\bar{q} = \frac{1}{2}\rho V_T^2 \tag{4.23}$$

in which ρ denotes the density of the air, and it varies along the altitude of the aircraft. Additionally, at certain Mach numbers and altitudes it is assumed that

$$F_T = C_{F\delta_t}\delta_t \tag{4.24}$$

where $C_{F\delta_t}$represents the maximum engine thrust the aircraft can provide, δ_t denotes the throttle position, which varies from 0 to 1, that is $0 \leq \delta_t \leq 1$. Therefore, combining equations (4.21), (4.22), (4.23) and (4.24), the longitudinal fixed-wing model can be rewritten as:

$$\begin{aligned}
m\dot{V}_T &= C_{F\delta_t}\delta_t\cos(\alpha+\alpha_T) - \frac{1}{2}\rho V_T^2 S C_D - mg\sin(\gamma) \\
m\dot{\gamma}V_T &= C_{F\delta_t}\delta_t\sin(\alpha+\alpha_T) + \frac{1}{2}\rho V_T^2 S C_L - mg\cos(\gamma) \\
\dot{\alpha} &= q - \dot{\gamma} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4.25) \\
\dot{q} &= \frac{\rho V_T^2 S\bar{c} C_m}{2I_y}
\end{aligned}$$

The modeling problem is now reduced to defining the lift, drag and pitching coefficients: C_D, C_L, and C_m, respectively. Normally, one may perform numerous experiments on a specific aircraft in the wind tunnel to determine these coefficients and set up a "lookup-table" in the aircraft computers since these coefficients vary according to different flight environments and it is difficult to determine explicit closed-form expressions for these variations. However, in certain defined flight environments, these behaviors will be simpler and can be expressed explicitly. In this project, it is assumed that the coefficients C_D, C_L, and C_m have the following explicit functional dependencies at certain Mach numbers and altitudes, that is:

$$\begin{aligned}
C_D &= C_D(\alpha) + C_{D\delta_e}\delta_e \qquad (4.26) \\
C_L &= C_L(\alpha) + C_{L\delta_e}\delta_e \qquad (4.27) \\
C_m &= C_m(\alpha) + \frac{\bar{c}}{2V_T}C_{mq}q + C_{m\delta_e}\delta_e \qquad (4.28)
\end{aligned}$$

where δ_e denotes the elevator angle, $C_{D\delta_e}$, $C_{L\delta_e}$, and $C_{m\delta_e}$ are all constant coefficients, and C_{mq} denotes pitch damping derivative. The functions $C_D(\alpha)$, $C_L(\alpha)$, and $C_m(\alpha)$ may be expressed explicitly according to the angle of attack α. Practically, using the experimentally generated "lookup-tables", one may perform polynomial regression in order to obtain these explicit functional dependencies. This is precisely what we have done in this project.

Polynomial regression for $C_D(\alpha)$

It is assumed that the following "lookup-table" characterizing $C_D(\alpha)$ for a certain UAV at the Mach number of 0.8 is available [52]:

Table 4.1: States of the UAV corresponding to a normal mode

Angle of attack (α)	$-2°$	$-1°$	$0°$	$1°$	$2°$	$3°$	$4°$	$5°$	$6°$	$7°$	$8°$	$9°$	$10°$
$C_D(\alpha)$	0.017	0.017	0.018	0.019	0.022	0.025	0.029	0.034	0.041	0.047	0.057	0.065	0.078

A number of polynomial regressions using different polynomial orders are performed as summarized in the following table (α in rad).

Table 4.2: Detection observer residual outputs corresponding to a normal mode

2^{nd} order	$f(\alpha) = 1.6545\alpha^2 + 0.0501\alpha + 0.0175$
	$\|C_D(\alpha) - f(\alpha)\|^2 = 6.8032 \times 10^{-6}$
3^{rd} order	$f(\alpha) = 2.4114\alpha^3 + 1.1495\alpha^2 + 0.067\alpha + 0.0179$
	$\|C_D(\alpha) - f(\alpha)\|^2 = 3.4186 \times 10^{-6}$
4^{th} order	$f(\alpha) = 12.3757\alpha^4 - 1.0445\alpha^3 + 1.3784\alpha^2 + 0.0687\alpha + 0.0177$
	$\|C_D(\alpha) - f(\alpha)\|^2 = 3.1548 \times 10^{-6}$
5^{th} order	$f(\alpha) = 221.1889\alpha^5 - 64.8338\alpha^4 + 6.7039\alpha^3 + 1.2608\alpha^2$ $+ 0.0586\alpha + 0.0178$
	$\|C_D(\alpha) - f(\alpha)\|^2 = 2.9214 \times 10^{-6}$

In this report, we are going to use the third order polynomial

$$C_D(\alpha) = 2.4114\alpha^3 + 1.1495\alpha^2 + 0.067\alpha + 0.0179 \tag{4.29}$$

where the Mach number is 0.8 and $-0.0349 \leq \alpha \leq 0.1745$.

Polynomial regression for $C_L(\alpha)$

It is assumed that the following "lookup-table" characterizing $C_L(\alpha)$ for a certain UAV at the Mach number of 0.8 is available:

Table 4.3: States of the UAV corresponding to a float fault in the elevator input channel

Angle of attack (α)	$-2°$	$-1°$	$0°$	$1°$	$2°$	$3°$	$4°$	$5°$	$6°$	$7°$	$8°$	$9°$	$10°$
$C_L(\alpha)$	0	0.045	0.095	0.14	0.19	0.23	0.28	0.325	0.37	0.42	0.46	0.50	0.54

A number of polynomial regressions using different polynomial orders are performed as shown in the table below (α in rad):

Table 4.4: Detection observer residual outputs corresponding

1^{st} order	$f(\alpha) = 2.6035\alpha + 0.0948$
	$\|C_L(\alpha) - f(\alpha)\|^2 = 2.3407 \times 10^{-4}$
2^{nd} order	$f(\alpha) = -0.9019\alpha^2 + 2.7294\alpha + 0.0942$
	$\|C_D(\alpha) - f(\alpha)\|^2 = 8.2967 \times 10^{-5}$

In this report, we are going to use the first order polynomial

$$C_L(\alpha) = 2.6035\alpha + 0.0948 \tag{4.30}$$

where the Mach number is 0.8 and $-0.0349 \leq \alpha \leq 0.1745$.

Polynomial regression for $C_m(\alpha)$

It is assumed that the following "lookup-table" regarding $C_m(\alpha)$ for a certain UAV operating at the Mach number of 0.8 is available:

Table 4.5: to float fault in the elevator input channel

Angle of attack (α)	$-2°$	$-1°$	$0°$	$1°$	$2°$	$3°$	$4°$	$5°$	$6°$	$7°$	$8°$	$9°$	$10°$
$C_m(\alpha)$	-0.01	-0.013	-0.016	-0.019	-0.022	-0.025	-0.028	-0.039	-0.05	-0.055	-0.06	-0.071	-0.082

A number of polynomial regressions using different polynomial orders are performed as summarized below (α in rad):

Table 4.6: Linear detection observer residual outputs corresponding

1^{st} order	$f(\alpha) = -1.2889\alpha^2 - 0.1588\alpha - 0.0148$
	$\|C_m(\alpha) - f(\alpha)\|^2 = 5.0773 \times 10^{-5}$
2^{nd} order	$f(\alpha) = 0.6577\alpha^3 - 1.4266\alpha^2 - 0.1542\alpha - 0.0147$
	$\|C_m(\alpha) - f(\alpha)\|^2 = 5.0521 \times 10^{-5}$
3^{rd} order	$f(\alpha) = 45.0696\alpha^4 - 11.9282\alpha^3 - 0.593\alpha^2 - 0.1479\alpha - 0.0154$
	$\|C_m(\alpha) - f(\alpha)\|^2 = 4.7023 \times 10^{-5}$

In this report, we are going to use the second order polynomial

$$C_m(\alpha) = -1.2889\alpha^2 - 0.1588\alpha - 0.0148 \tag{4.31}$$

where the Mach number is 0.8 and $-0.0349 \leq \alpha \leq 0.1745$.

Given the above explicit functional dependencies for the coefficients $C_{D\delta_e}$, $C_{L\delta_e}$, and $C_{m\delta_e}$, by combining equations (5-21)-(5-26), the nonlinear longitudinal fixed-wing UAV model may now be written as follows:

$$\begin{aligned}
m\dot{V}_T &= C_{F\delta_t}\delta_t\cos(\alpha+\alpha_T) - \frac{1}{2}\rho V_T^2 S\left(2.4114\alpha^3 + 1.1495\alpha^2 + 0.067\alpha + 0.0179 + C_{D\delta_e}\delta_e\right) \\
&\quad - mg\sin(\gamma) \\
m\dot{\gamma}V_T &= C_{F\delta_t}\delta_t\sin(\alpha+\alpha_T) + \frac{1}{2}\rho V_T^2 S\left(2.6035\alpha + 0.0948 + C_{L\delta_e}\delta_e\right) - mg\cos(\gamma) \\
\dot{\alpha} &= q - \dot{\gamma} \\
\dot{q} &= \frac{1}{2I_y}\rho V_T^2 S\bar{c}\left(-1.2889\alpha^2 - 0.1588\alpha - 0.0148 + \frac{\bar{c}}{2V_T}C_{mq}q + C_{m\delta_e}\delta_e\right)
\end{aligned} \tag{4.32}$$

where the throttle position δ_t and the elevator angle δ_e are the only two constraint inputs, namely $0 \leq \delta_t \leq 1$ and $-25° \leq \delta_e \leq 25°$. In addition, we assume that all the states are available for measurement:

$$\begin{aligned} y_1 &= V_T \\ y_2 &= \gamma \\ y_3 &= \alpha \\ y_4 &= q \end{aligned} \tag{4.33}$$

Next, we will define the fault modes in the dynamical model. We assume that two inputs δ_t and δ_e could both have possible anomalies which are denoted by δ_t^f and δ_e^f in the input channels. Also, it is assumed that the UAV has two redundant inputs, namely δ_t^r and δ_e^r, so that equation (4.32) can be re-written as:

$$\begin{aligned} m\dot{V}_T &= C_{F\delta_t}(\delta_t + \delta_t^f + \delta_t^r)\cos(\alpha + \alpha_T) \\ &\quad - \frac{1}{2}\rho V_T^2 S\left(2.4114\alpha^3 + 1.1495\alpha^2 + 0.067\alpha + 0.0179 + C_{D\delta_e}(\delta_e + \delta_e^f + \delta_e^r)\right) - mg\sin(\gamma) \\ m\dot{\gamma}V_T &= C_{F\delta_t}(\delta_t + \delta_t^f + \delta_t^r)\sin(\alpha + \alpha_T) \\ &\quad + \frac{1}{2}\rho V_T^2 S\left(2.6035\alpha + 0.0948 + C_{L\delta_e}(\delta_e + \delta_e^f + \delta_e^r)\right) - mg\cos(\gamma) \\ \dot{\alpha} &= q - \dot{\gamma} \\ \dot{q} &= \frac{1}{2I_y}\rho V_T^2 S\bar{c}\left(-1.2889\alpha^2 - 0.1588\alpha - 0.0148 + \frac{\bar{c}}{2V_T}C_{mq}q + C_{m\delta_e}(\delta_e + \delta_e^f + \delta_e^r)\right) \end{aligned} \tag{4.34}$$

Moreover, it is assumed that $\delta_t^f = -k_t\delta_t$ and $\delta_e^f = -k_e\delta_e$, where k_t and k_e are piecewise constant, $0 \leq k_t, k_e \leq 1$. When $k_t, k_e = 1$, it represents the scenario of complete fault, whereas when $k_t, k_e = 0$, it corresponds to the healthy scenario, in which the redundant inputs δ_t^r and δ_e^r will be set to zero.

Linearization of the longitudinal fixed-wing model

In this section, we will perform linearization for the longitudinal fixed-wing model at a certain operating point. Toward this end, we need to specify all the parameters in equation (4.34), which are subsequently used in our numerical simulations. The table below summarizes these values:

4.2.2 Fault Detection and Isolation for the Input Channel δ_e

In this subsection, we will design an FDIR scheme for the input channel δ_e based on the nonlinear fixed-wing UAV model (4.34).

Application of the Geometric Approach

The purpose of applying a geometric approach [49, 50] is to find a new set of states which are affected by the input channel δ_t and are decoupled from the input channel δ_e, , through which one may design a residual generator for the actuator fault δ_t^f.

Table 4.7: to float fault in the elevator input channel

Engine thrust and the fuselage reference line angle	$\alpha_T = 0$
Wing platform area	$S = 300 ft^2 = 27.8709 m^2$
Density of air	$\rho = 8.9068 \times 10^{-4} slugs/ft^3 = 0.4590 kg/m^3$(h = 3000 ft)
Mean aerodynamic chord	$\bar{c} = 11.32$ft $= 3.4503$m
$y-$body axis moment of inertia	$I_y = 55814$slugs $\cdot ft^2 = 75673 kg \cdot m^2$
Aircraft total mass	$m = 20500$lbs $= 9298.6$kg
Acceleration due to gravity	$g = 32.2 ft/s^2 = 9.80665 m/s^2$
Elevator coefficient in drag	$C_{D\delta_e} = -4.0268$
Elevator coefficient in lift	$C_{L\delta_e} = -5.8780$
Elevator coefficient in pitch moment	$C_{m\delta_e} = -0.83232$
Engine thrust coefficient	$C_{F\delta_t} = 23422$lbs $= 104190$N
Pitch damping derivative	$C_{mq} = -32$

In this subsection, let us denote $x = \left[\begin{array}{cccc} V_T & \gamma & \alpha & q \end{array}\right]^T$then equation (5-32) can be rewritten as

$$\left[\begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{array}\right] = \left[\begin{array}{l} -\frac{1}{2}\frac{\rho x_1^2 S}{m}\left(2.4114x_3^3 + 1.1495x_3^2 + 0.067x_3 + 0.0179\right) - g\sin(x_2) \\ \frac{1}{2}\frac{\rho x_1 S}{m}(2.6035x_3 + 0.0948) - \frac{g}{x_1}\cos(x_2) \\ x_4 - \frac{1}{2}\frac{\rho x_1 S}{m}\left(2.6035x_3 + 0.0948\right) + \frac{g}{x_1}\cos(x_2) \\ \frac{1}{2I_y}\rho x_1^2 S\bar{c}\left(-1.2889x_3^2 - 0.1588x_3 - 0.0148 + \frac{\bar{c}}{2V_T}C_{mq}x_4\right) \end{array}\right]$$
$$+ \left[\begin{array}{l} \frac{C_{F\delta_t}\cos(x_3)}{m} \\ \frac{C_{F\delta_t}\sin(x_3)}{mx_1} \\ -\frac{C_{F\delta_t}\sin(x_3)}{mx_1} \\ 0 \end{array}\right]\left(\delta_t + \delta_t^f\right) + \left[\begin{array}{l} -\frac{1}{2}\frac{\rho x_1^2 S}{m}C_{D\delta_e} \\ \frac{1}{2}\frac{\rho x_1 S}{m}C_{L\delta_e} \\ -\frac{1}{2}\frac{\rho x_1 S}{m}C_{L\delta_e} \\ \frac{1}{2I_y}\rho x_1^2 S\bar{c}C_{m\delta_e} \end{array}\right]\left(\delta_e + \delta_e^f\right)$$

and the output equation is

$$y = \left[\begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array}\right]^T$$

If we just consider the state space equation for x_4 we have

$$\dot{x}_4 = \frac{1}{2I_y}\rho y_1^2 S\bar{c}\left(-1.2889y_3^2 - 0.1588y_3 - 0.0148 + \frac{\bar{c}}{2V_T}C_{mq}x_4\right) + \frac{1}{2I_y}\rho y_1^2 S\bar{c}C_{m\delta_e}(\delta_e + \delta_e^f)$$

Therefore x_4 is only affected by $\delta_e$and is decoupled from δ_t and we need just to design an observer for x_4 in order to detect faults in input channelδ_e. Let $e = x_4 - \hat{x}_4$ and

$\dot{\hat{x}}_4 = \frac{1}{2I_y}\rho y_1^2 S\bar{c}\left(-1.2889y_3^2 - 0.1588y_3 - 0.0148 + \frac{\bar{c}}{2V_T}C_{mq}y_4\right) + \frac{1}{2I_y}\rho y_1^2 S\bar{c}C_{m\delta_e}(\delta_e) + k(x_4 - \hat{x}_4)$where $k > 0$. Then we have

$$\begin{aligned} \dot{e} &= -ke + \tfrac{1}{2I_y}\rho y_1^2 S\bar{c}C_{m\delta_e}\delta_e^f \\ r &= e \end{aligned}$$

Therefore if there is no fault in input channelδ_e, the residual signal $r$goes to zero, but if there is a fault in δ_e, the residual signal r will be affected by this fault.

4.2.3 Fault Detection and Isolation for the Input Channel δ_t

In this subsection, we will apply the same FDIR scheme proposed for the input channel δ_e to the second input channelδ_t.

In this case, the purpose of applying the geometric approach is to find a new set of states which are affected by the signals from the input channel δ_t and are decoupled from the input channel δ_e, through which one may design a residual generator for the actuator fault δ_t^f.

In this subsection, let us denote

$$l(x) = \begin{bmatrix} \frac{C_{F\delta_t}\cos(x_3+\alpha_T)}{m} & \frac{C_{F\delta_t}\sin(x_3+\alpha_T)}{mx_1} & -\frac{C_{F\delta_t}\sin(x_3+\alpha_T)}{mx_1} & 0 \end{bmatrix}^T$$

and

$$p(x) = \begin{bmatrix} -\frac{1}{2}\frac{\rho x_1^2 S}{m}C_{D\delta_e} & \frac{1}{2}\frac{\rho x_1 S}{m}C_{L\delta_e} & -\frac{1}{2}\frac{\rho x_1 S}{m}C_{L\delta_e} & \frac{1}{2I_y}\rho x_1^2 S\bar{c}C_{m\delta_e} \end{bmatrix}^T$$

First we find the conditioned invariant distribution $\sum_*^P$ that containing $P = span\{p(x)\}$ and involutive. Since we have full state measurement $(h(x) = x)$, every distribution is conditioned invariant and therefore

$$\sum\nolimits_*^P = p(x)$$

Next, by using observability distribution algorithm, we have

$$\Omega^* = o.c.a.\left(\left(\sum\nolimits_*^P\right)^\perp\right) = \left(\sum\nolimits_*^P\right)^\perp = P^\perp$$

The necessary condition for solving the nonlinear fundamental problem of residual generation (defined in [49]) is:

$$(span\,\{l\})^\perp + \text{o.c.a.}\left(\left(\sum\nolimits_*^P\right)^\perp\right) = T^*X$$

which is satisfied since

$$(span\,\{l\}) \cap (span\{p\}) = 0$$

since $P = span\,\{p(x)\}$ is involutive, Ω^* can be spanned by exact differentials as follow:

$$\Omega^* = span\left\{d(\frac{\bar{c}}{I_y}C_{m\delta_e}x_1 + \frac{1}{m}C_{D\delta_e}x_4), d(x_2 + x_3), d(\frac{1}{C_{D\delta_e}}\log x_1 + \frac{1}{C_{L\delta_e}}x_2)\right\}$$

The dimension of the states in new transformation that are affected by $l(x)$ and is decoupled from $p(x)$ is the dimension of Ω^*, therefore in order to find the minimum order observer , we should find the smallest Ω^*which satisfied above conditions but there is no straight algorithm for finding this codistribution.

In fixed-wing model since we have the full state measurement every codistribution is an observability codistribution for our model. Therefore we can consider the distribution $P = span\{p(x), p_1(x), p_2(x)\}$ where $p_1(x) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$and $p_2(x) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$.

In this case $\Omega^* = P^\perp$ and since $P = span\{p(x)\}$ is involutive, Ω^* can be spanned by exact differentials as follow:

$$\Omega^* = span\left\{d(\frac{1}{C_{D\delta_e}}\log x_1 + \frac{1}{C_{L\delta_e}}x_2)\right\}$$

We have

$$\Omega^* \cap span\{dh\} = \Omega^* = span\{d(\Psi_1 \circ h)\} = span\{d(\Psi_1)\}$$

where $\Psi_1 : R^4 \to R^1$

$$\Psi_1(x) = \frac{1}{C_{D\delta_e}}\log x_1 + \frac{1}{C_{L\delta_e}}x_2$$

Let

$$H_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

then

$$\Psi(y) = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \Psi_1(y) \\ H_2 y \end{pmatrix}$$

is a diffeomorphism in output space and

$$\Phi(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \Phi_1(x) \\ H_2 h(x) \end{pmatrix}$$

is a diffeomorphism in state space.

We can now apply this diffeomorphism as the coordinate transformation $z = \Phi(x)$ and find the new state space equation for$z$.

We have

$$z_1 = \tfrac{1}{C_{D\delta_e}}\log x_1 + \tfrac{1}{C_{L\delta_e}}x_2,$$
$$z_2 = x_2 \quad z_3 = x_3 \quad z_4 = x_4$$

and

$$x = \Phi^{-1}(z) = \begin{bmatrix} \exp(C_{D\delta_e}(z_1 - \frac{1}{C_{L\delta_e}}z_2) \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

$z_1$is the only state that we need for fault detection in input channel δ_t and

$$\dot{z}_1 = -\tfrac{1}{2}\tfrac{\rho S}{m}\exp\left(C_{D\delta_e}(z_1 - \tfrac{1}{C_{L\delta_e}}z_2)\right)\left[\tfrac{2.4114z_3^3}{C_{D\delta_e}} + \tfrac{1.1495z_3^2}{C_{D\delta_e}} + \tfrac{0.067z_3}{C_{D\delta_e}} + \tfrac{0.0179}{C_{D\delta_e}} - \tfrac{2.6035z_3}{C_{L\delta_e}} - \tfrac{0.0948}{C_{L\delta_e}}\right] -$$
$$-g\exp\left(C_{D\delta_e}(-z_1 + \tfrac{1}{C_{L\delta_e}}z_2)\right)\left(\tfrac{\sin(z_2)}{C_{D\delta_e}} + \tfrac{\cos(z_2)}{C_{L\delta_e}}\right) +$$
$$\tfrac{C_{F\delta_t}}{m}\exp\left(C_{D\delta_e}(-z_1 + \tfrac{1}{C_{L\delta_e}}z_2)\right)\left(\tfrac{\cos(z_3+\alpha_T)}{C_{D\delta_e}} + \tfrac{\sin(z_3+\alpha_T)}{C_{L\delta_e}}\right)\delta_t$$

The output space and state space diffeomorphisms are identical therefore in new coordinates we have$y = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}$. We can rewrite the above equation as

$$\dot{z}_1 = f(z_1, y_2, y_3) + g(z_1, y_2, y_3)\delta_t$$

The observer for z_1 can be designed as

$$\dot{\hat{z}}_1 = f(z_1, y_2, y_3) + g(z_1, y_2, y_3)\delta_t + k(\hat{z}_1 - z_1)$$

where $k > 0$.

4.2.4 Simulation Results of FDI Scheme for the Longitudinal Fixed Wing UAV Model

In this section, simulation results for our proposed nonlinear FDI scheme applied to the nonlinear longitudinal fixed wing UAV model will be presented. Different actuator faults have been considered in elevator and throttle input channel. The initial conditions of the simulation system are selected as:

$$V_T^0 = 282m/s\ ,\ \gamma^0 = 1^\circ = 0.0175rad\ ,\ \alpha^0 = -0.1^\circ = -0.00175rad,\ and\ q^0 = 0\ rad/s$$

The model predictive controller has been designed for stabilizing the UAV and reference trajectory signals are selected as:

$$V_T^0 = 272m/s\ ,\ \gamma^0 = 0^\circ,\ \alpha^0 = -0.1^\circ = -0.00175rad,\ and\ q^0 = 0\ rad/s$$

Figures 4.2 and 4.3 depict the states of the UAV and detection observer residual outputs when there is no fault in the system.

Figures 4.4 and 4.5 depict the states and residual outputs when there is a float fault in the elevator input channel at t=20sec. As it is shown in these figures, the UAV is become unstable in this fault scenario.

In order to show the effectiveness of nonlinear FDI in comparison to linear FDI, a linear detection filter was designed for the linearized model of UAV. Figure 4.6 shows the residual outputs of linear detection observer when there is a float fault in the elevator channel and in this case linear observer cannot isolate this fault since both residuals have been affected by float fault in the elevator channel but the nonlinear detection filter perfectly detects and isolates this fault.

Figures 4.7 and 4.8 depict the states and residual outputs when there is a Hard over fault in the elevator input channel at t=20sec. The UAV becomes unstable in less than 1 second after the occurrence of hard over fault.

Figures 4.9 and 4.10 show the states and residual outputs when there is a loss of effectiveness fault (k=0.5) in the elevator input channel at t=20sec. The UAV remains stable but it does not converge to reference setpoints.

Figures 4.11 and 4.12 depict the states and residual outputs when there is a loss of effectiveness fault (k=0.5) in the elevator input channel at t=20sec. The UAV remains stable but it does not converge to reference setpoints.

Figures 4.13 and 4.14 depict the states and residual outputs when there is a loss of effectiveness fault (k=0.8) in the elevator input channel at t=20sec. The UAV remains stable but it does not converge to reference setpoints.

Figures 4.15 and 4.16 show the states and residual outputs when there is a float fault in throttle input channel at t=25sec. Figure 4.17 depicts the residual outputs of linear observer. As it is shown in this figure we cannot isolate the faults completely by linear detection observer.

Figures 4.18 and 4.19 show the states and residual outputs when there is a hard over fault in throttle input channel at t=25sec.

Figures 4.20 and 4.21 depict the states and residual outputs when there is a loss of effectiveness fault (k=0.2) in throttle input channel at t=25sec.

Figures 4.22 and 4.23 depict the states and residual outputs when there is a loss of effectiveness fault (k=0.5) in throttle input channel at t=25sec.

Figures 4.24 and 4.25 depict the states and residual outputs when there is a loss of effectiveness fault (k=0.8) in the throttle input channel at t=25sec.

In all above figures, the throttle residual output is completely affected by the faults in the throttle input channel and elevator residual output is completely decoupled from these faults.

Finally Figures 4.26 and 4.27 depict the states and residual outputs when there are simultaneous faults in both input channel. As it is shown in figure 4.27 the residual outputs perfectly detect and isolate both faults.

Based on the 5% disturbance in both input channels, we choose the residual fault threshold to 1×10^{-3} for both linear and nonlinear elevator detection observers and 1×10^{-4} for throttle detection observer. Following table summarizes the simulation result for both linear and nonlinear detection observers.

As it is shown in the above table, linear observer cannot detect 10% loss of effectiveness in the elevator channel and it also produces the false alarm in throttle residual when there is a float fault in elevator channel.

Table 4.8: to a loss of effectiveness (k=0.8) in the throttle input channel

Input Channel	Fault type	Nonlinear observer		Linear Observer	
		Elevator Residual	Throttle Residual	Elevator Residual	Throttle Residual
Elevator	Float	Detected	-	Detected	False Alarm
Elevator	Loss of effectiveness (k=0.9)	Detected	-	Not Detected	-
Elevator	Loss of effectiveness (k=0.8)	Detected	-	Detected	-
Elevator	Loss of effectiveness (k=0.5)	Detected	-	Detected	-
Throttle	Float	-	Detected	-	Detected
Throttle	Hard over	-	Detected	-	Detected
Throttle	Loss of effectiveness (k=0.5)	-	Detected	-	Detected
Throttle	Loss of effectiveness (k=0.8)	-	Detected	-	Detected
Throttle	Loss of effectiveness (k=0.9)	-	Detected	-	Detected

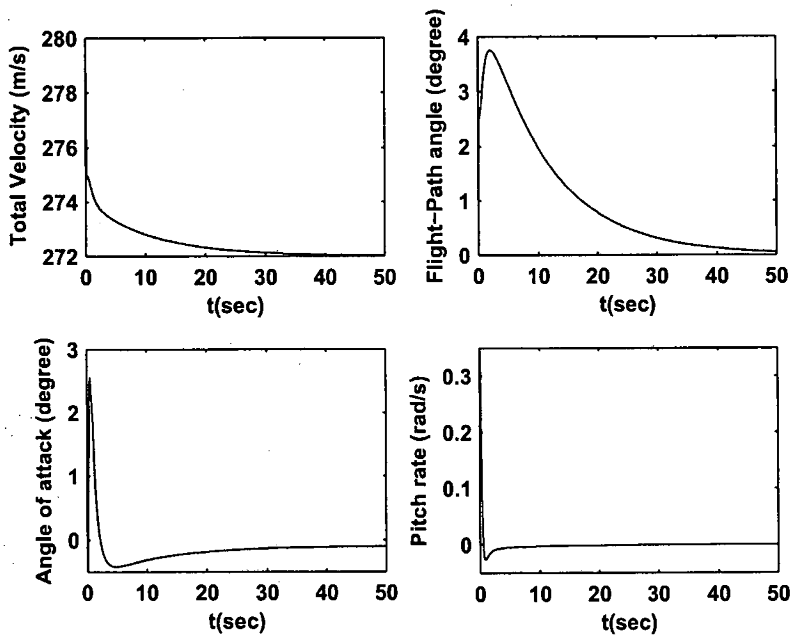


Figure 4.2: States of the UAV corresponding to a normal mode

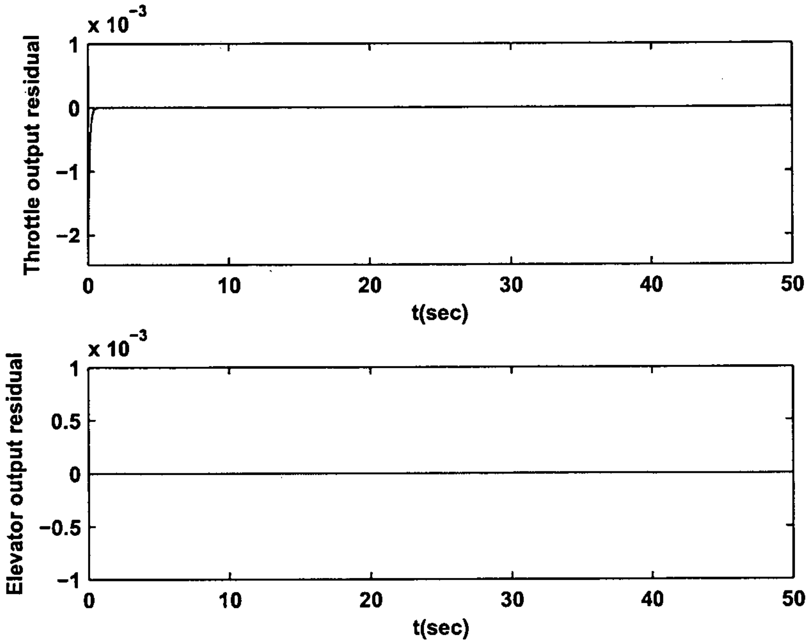


Figure 4.3: Detection observer residual outputs corresponding to a normal mode

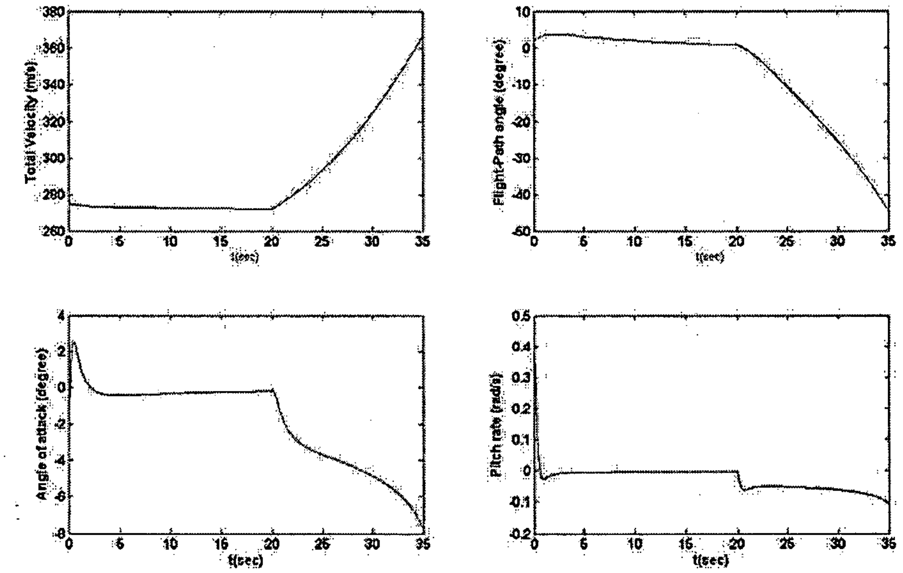


Figure 4.4: States of the UAV corresponding to a float fault in the elevator input channel

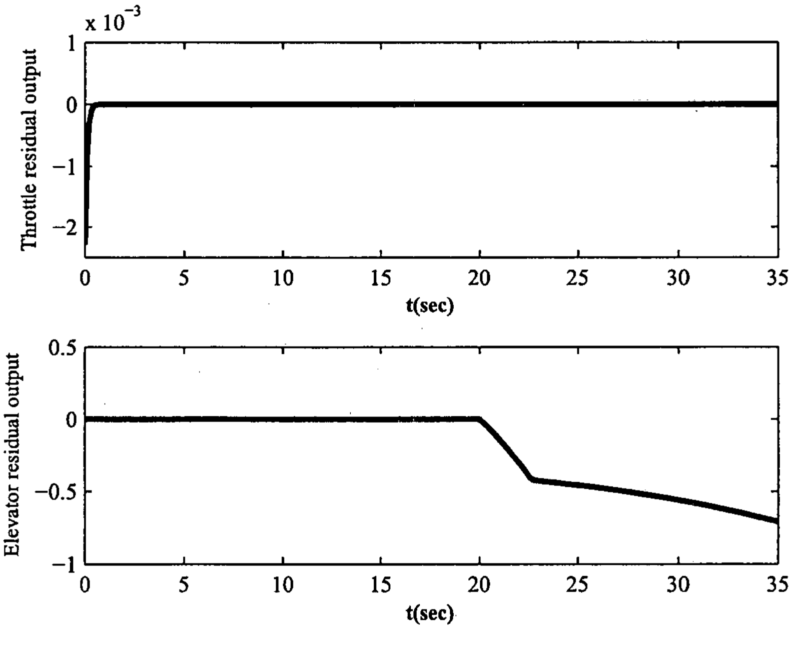


Figure 4.5: Detection observer residual outputs corresponding to float fault in the elevator input channel

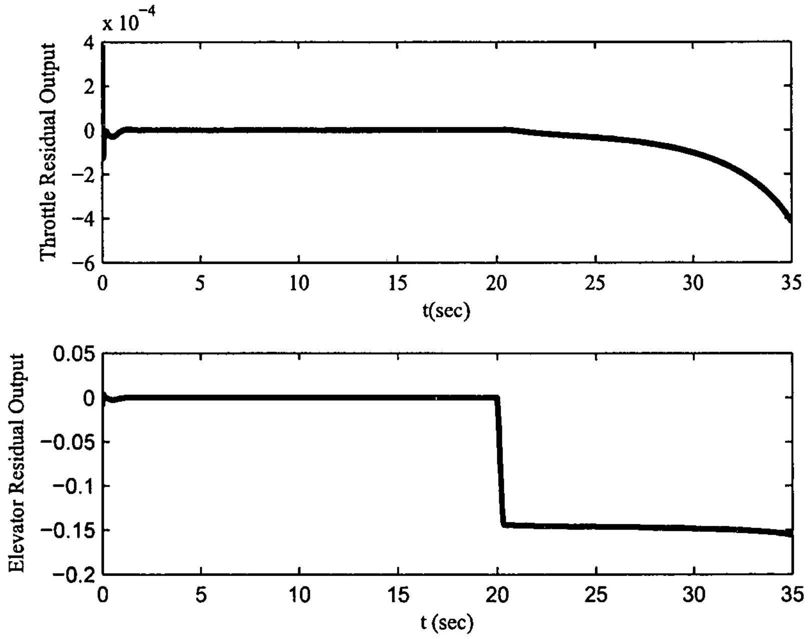


Figure 4.6: Linear detection observer residual outputs corresponding to float fault in the elevator input channel

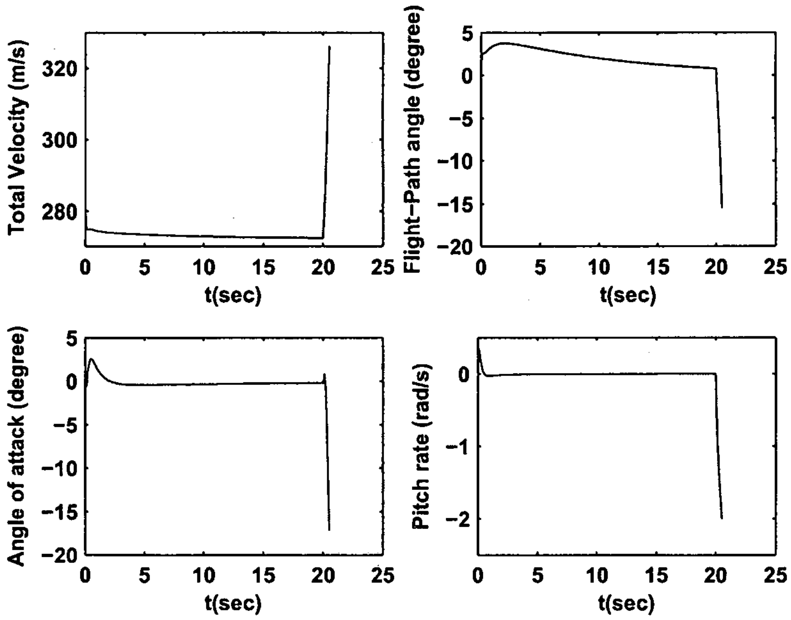


Figure 4.7: States of the UAV corresponding to a Hard over fault in the elevator input channel

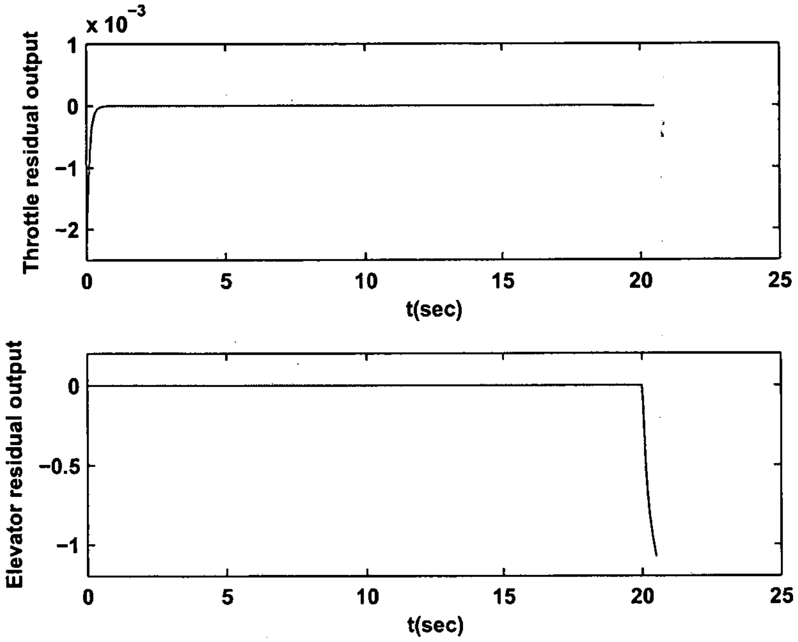


Figure 4.8: Detection observer residual outputs corresponding to Hard over in the elevator input channel

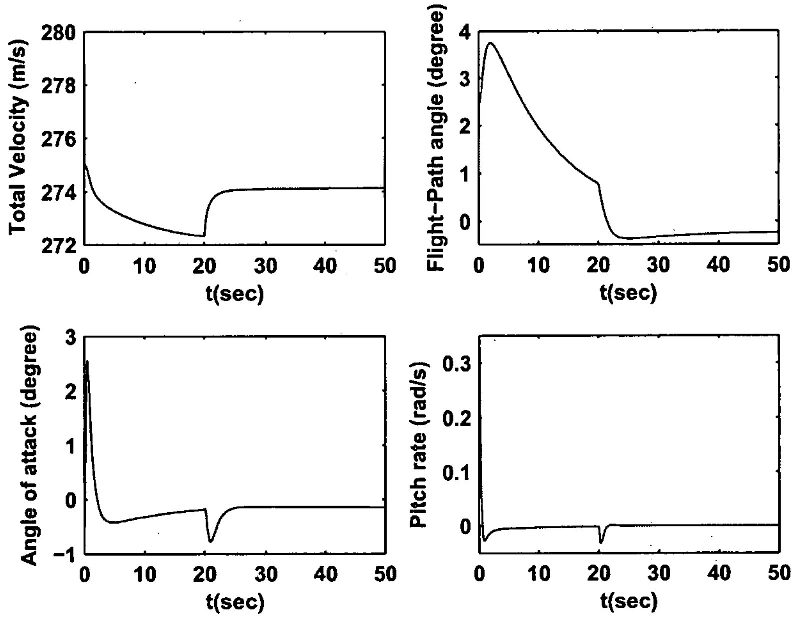


Figure 4.9: States of the UAV corresponding to a loss of effectiveness (k=0.2) fault in the elevator input channel

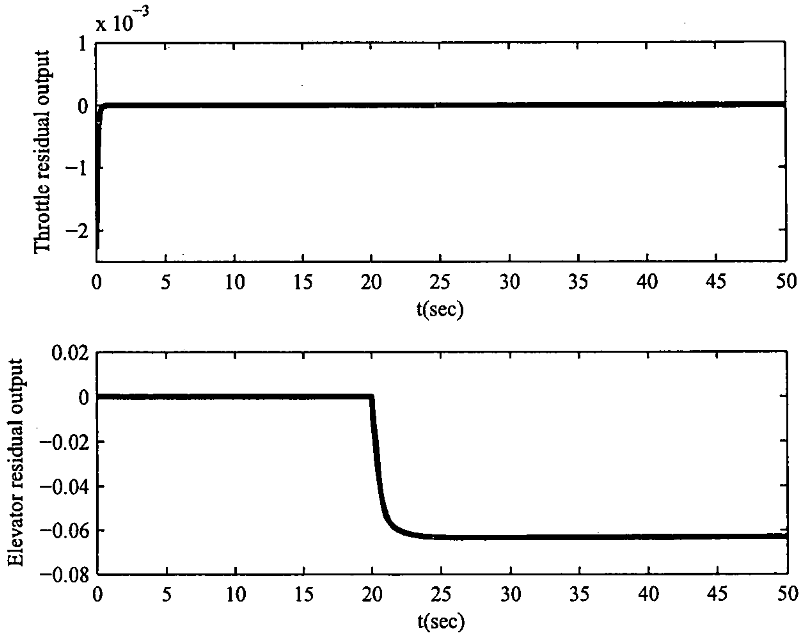


Figure 4.10: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.2) fault in the elevator input channel

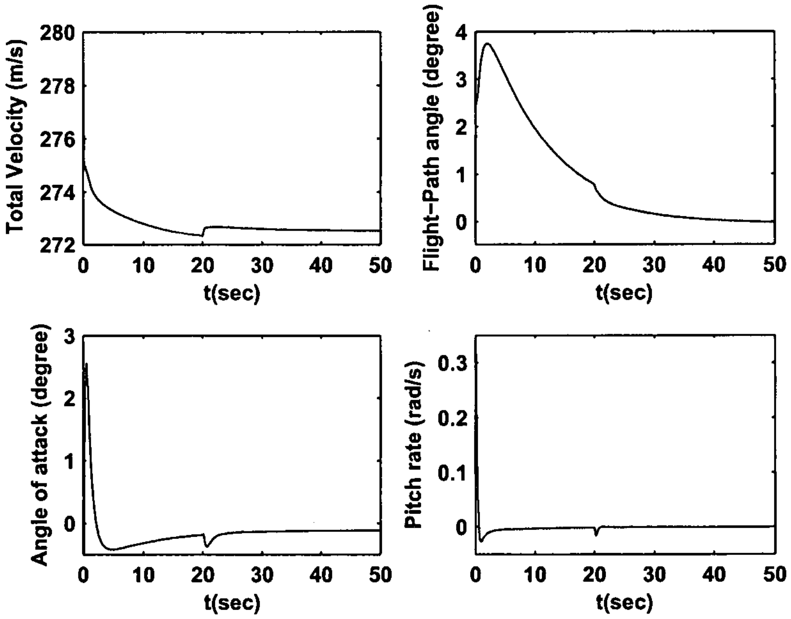


Figure 4.11: States of the UAV corresponding to a loss of effectiveness (k=0.5) fault in the elevator input channel

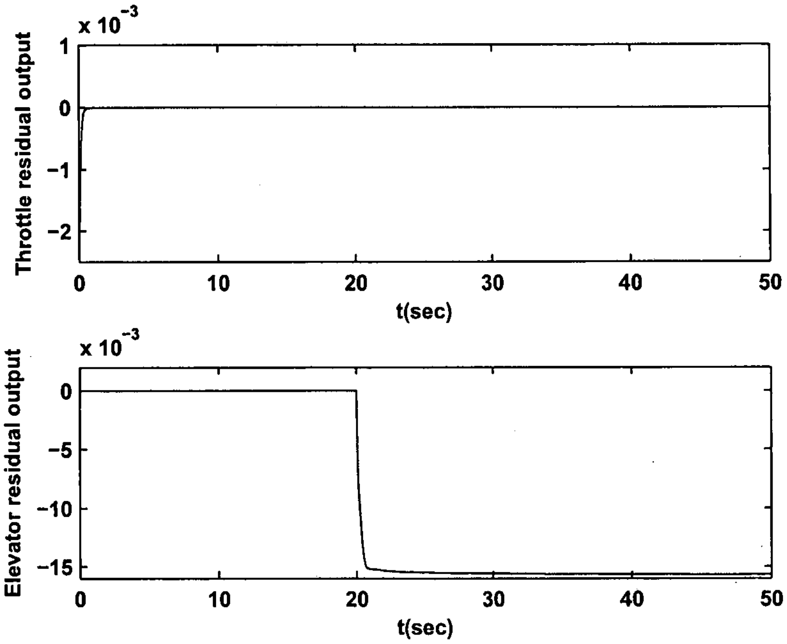


Figure 4.12: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.5) fault in the elevator input channel

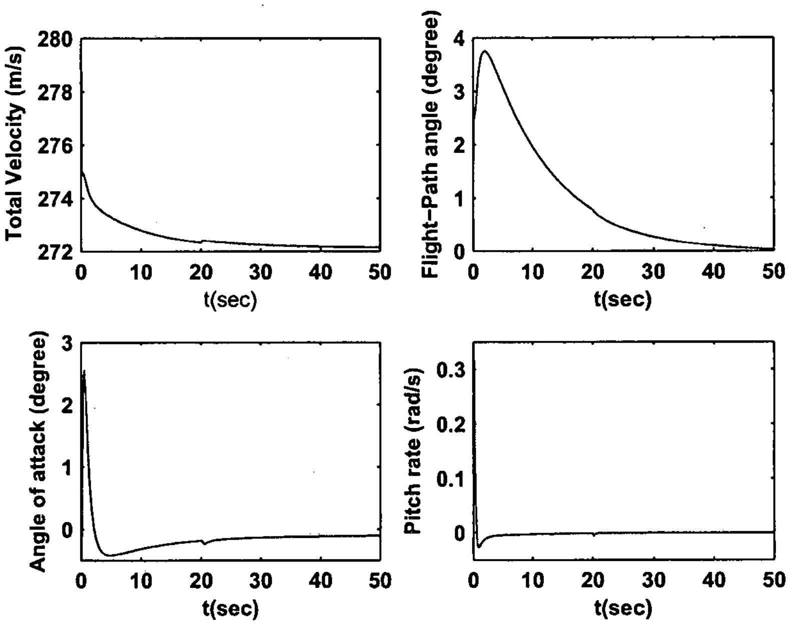


Figure 4.13: States of the UAV corresponding to a loss of effectiveness (k=0.8) fault in the elevator input channel

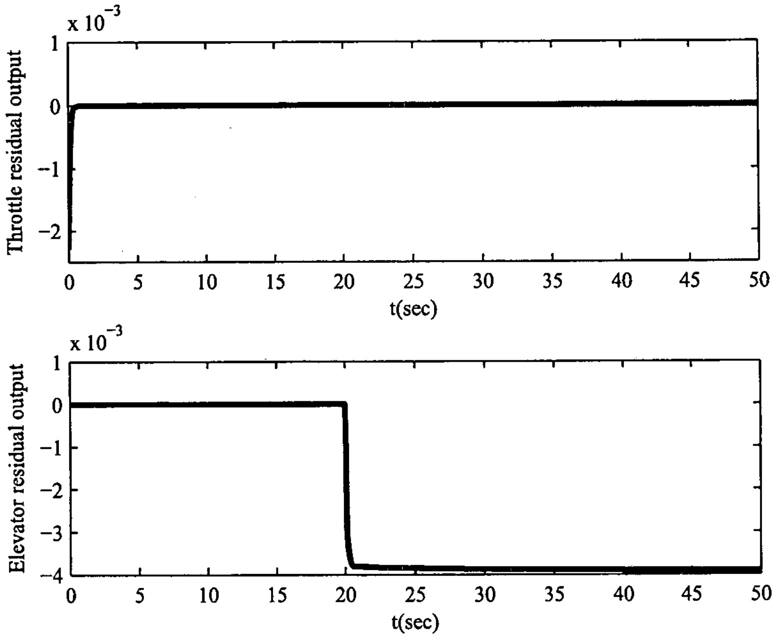


Figure 4.14: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.8) fault in the elevator input channel

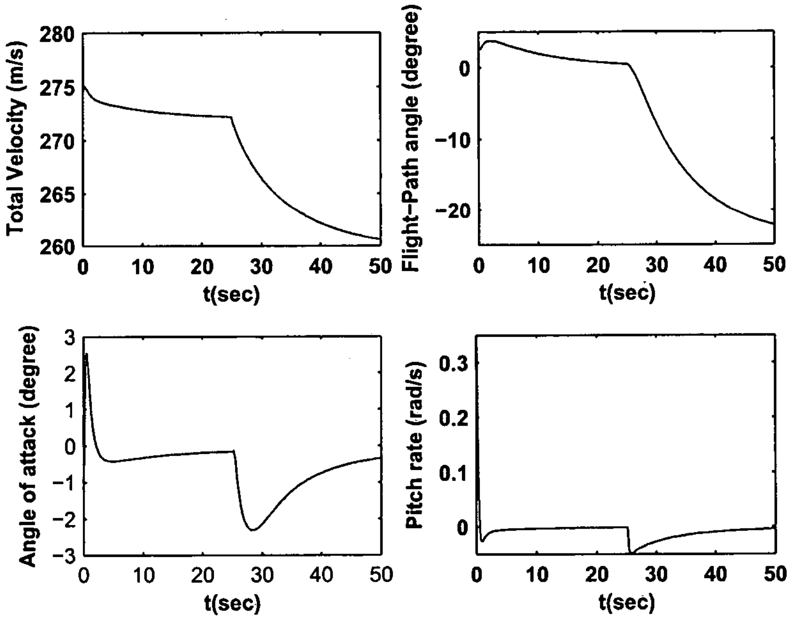


Figure 4.15: States of the UAV corresponding to a float fault in throttle input channel

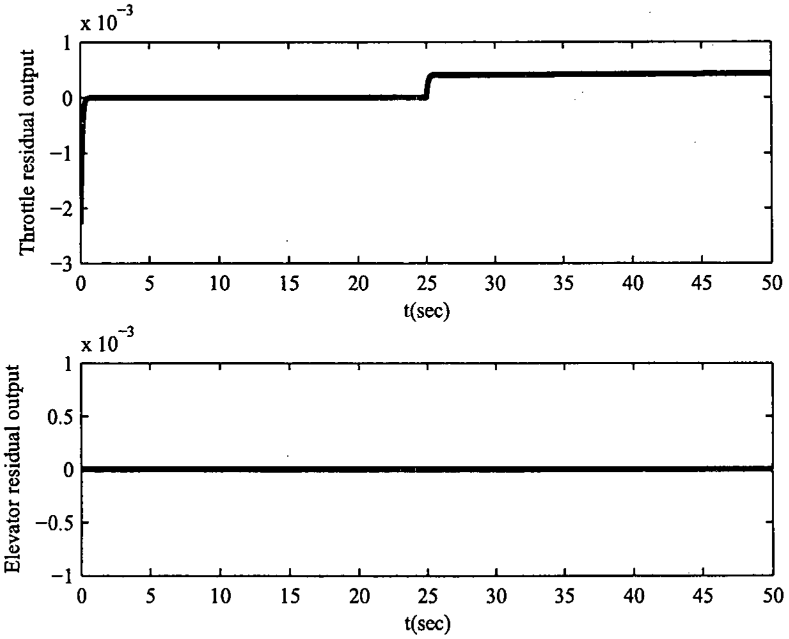


Figure 4.16: Nonlinear detection observer residual outputs corresponding to a float fault in throttle input channel

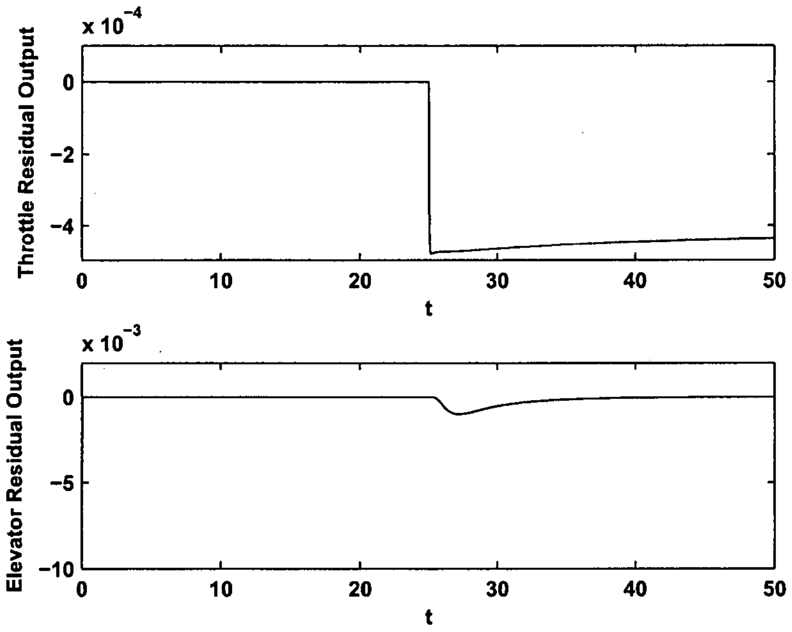


Figure 4.17: Linear detection observer residual outputs corresponding to a float fault in throttle input channel

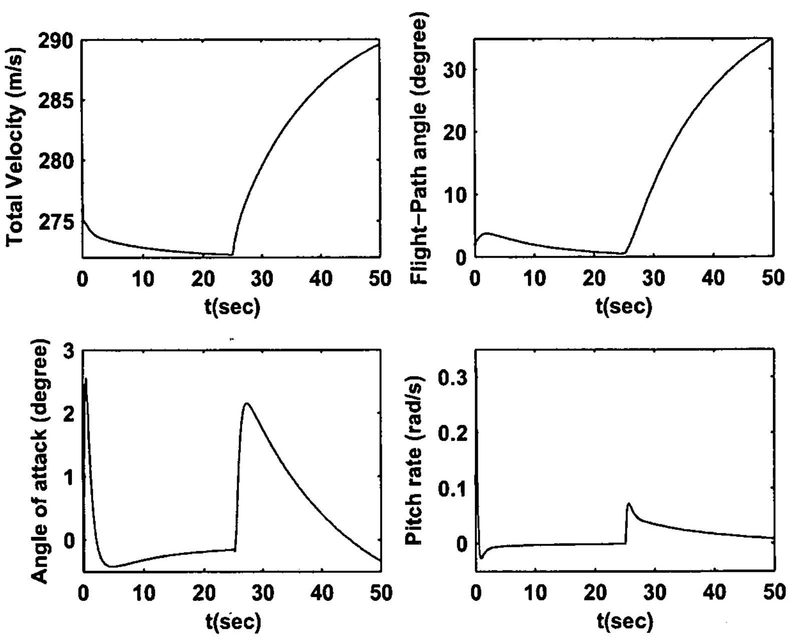


Figure 4.18: States of the UAV corresponding to a Hard over fault in throttle input channel

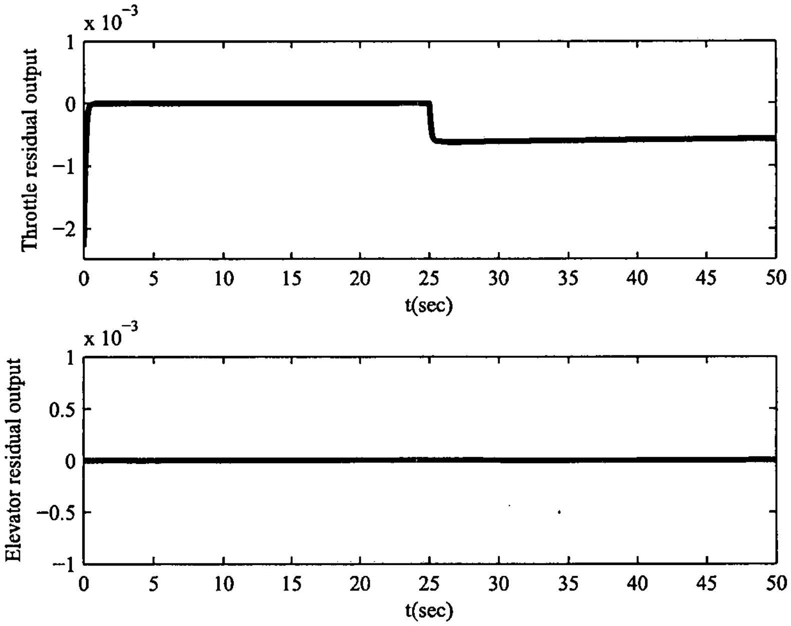


Figure 4.19: Detection observer residual outputs corresponding to a Hard over fault in throttle input channel

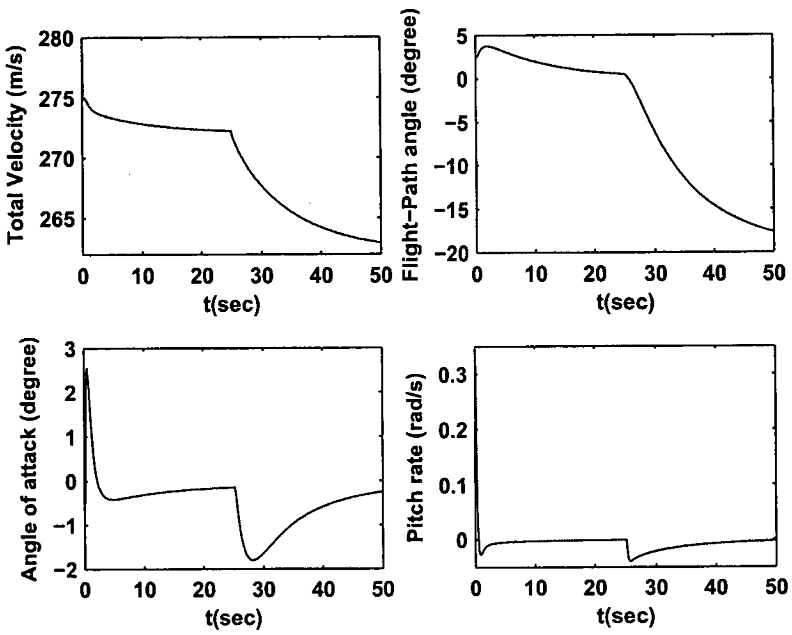


Figure 4.20: States of the UAV corresponding to a loss of effectiveness fault (k=0.2) in throttle input channel

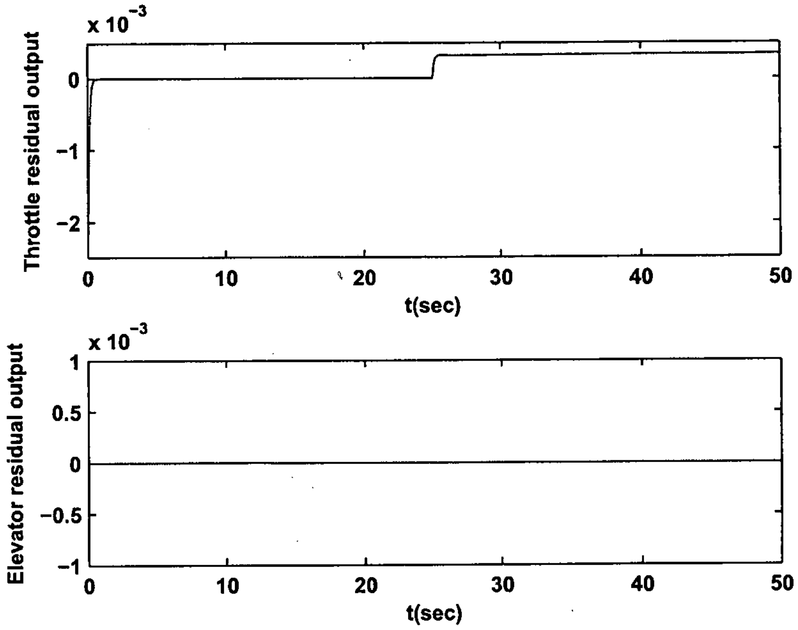


Figure 4.21: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.2) in throttle input channel

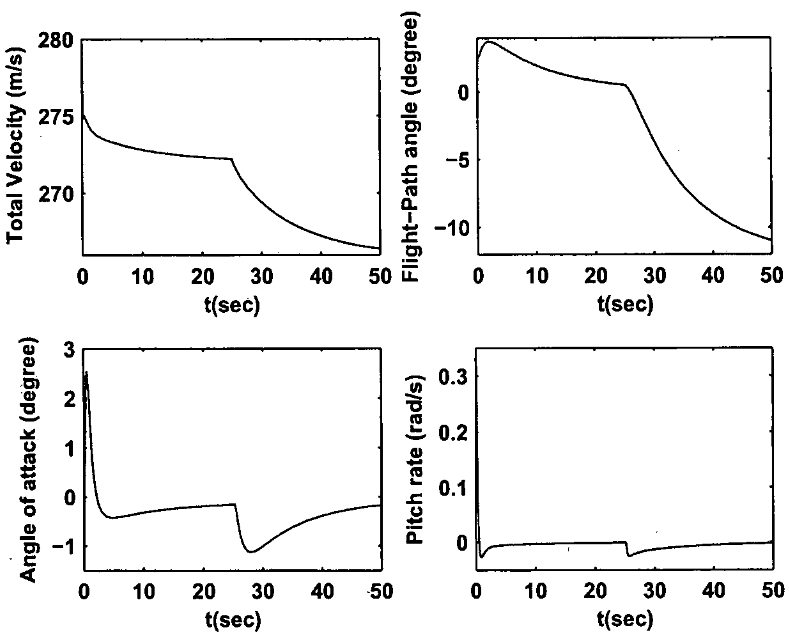


Figure 4.22: States of the UAV corresponding to a loss of effectiveness fault (k=0.5) in throttle input channel

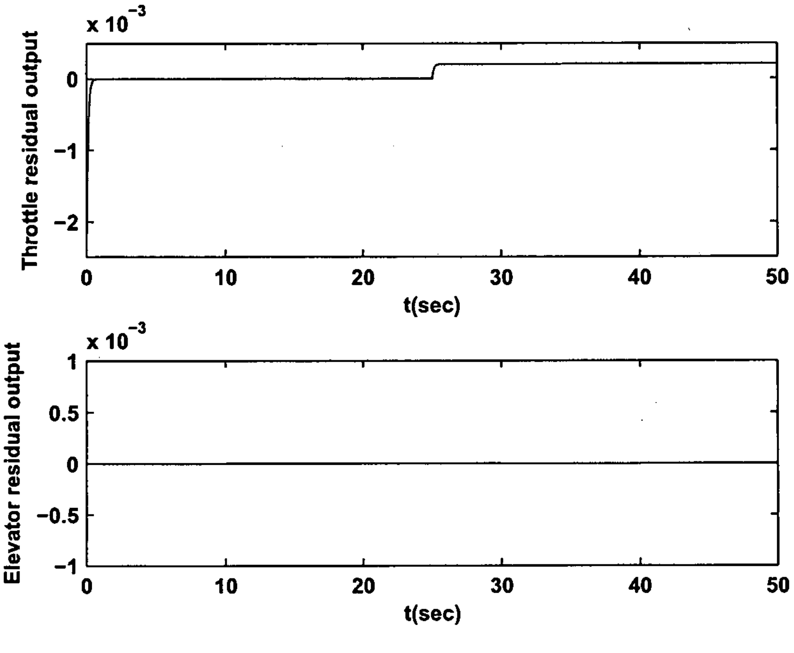


Figure 4.23: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.5) in throttle input channel

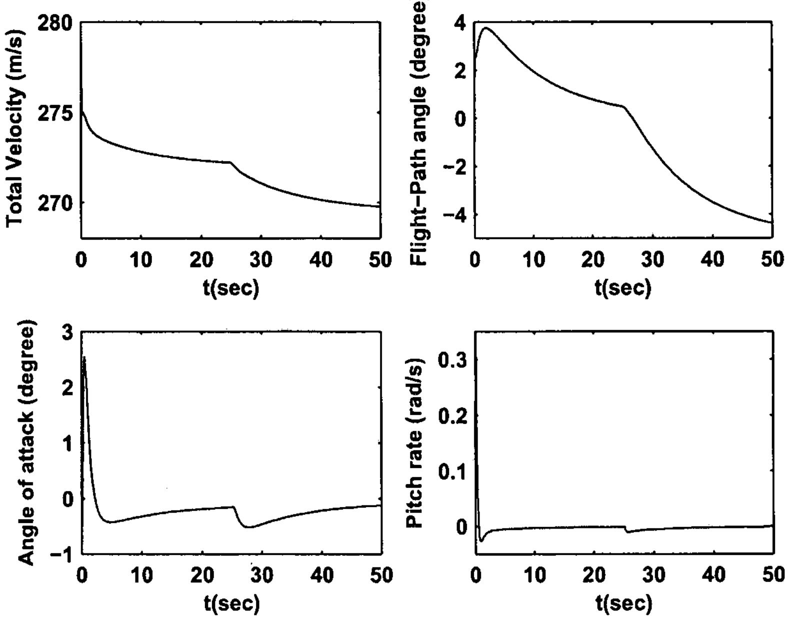


Figure 4.24: States of the UAV corresponding to a loss of effectiveness fault (k=0.8) in the throttle input channel

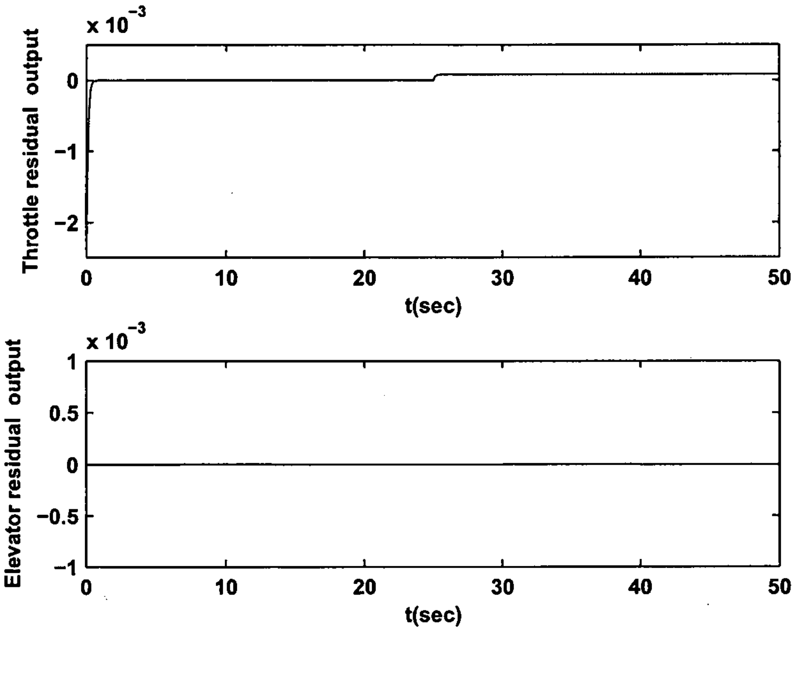


Figure 4.25: Detection observer residual outputs corresponding to a loss of effectiveness (k=0.8) in the throttle input channel

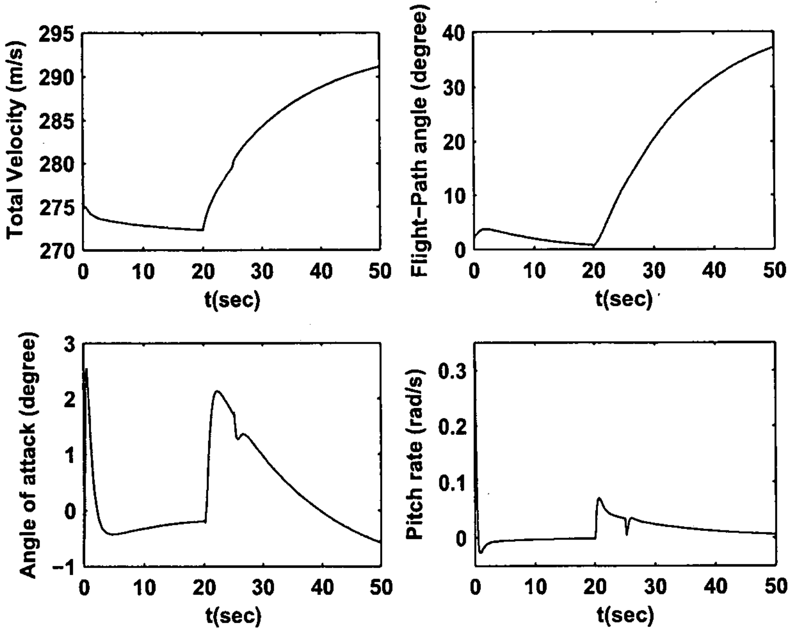


Figure 4.26: States of UAV corresponding to simultaneous faults in both input channel: Hard over fault in the throttle input at t=20 and Loss of performance (k=0.5) in the elevator input

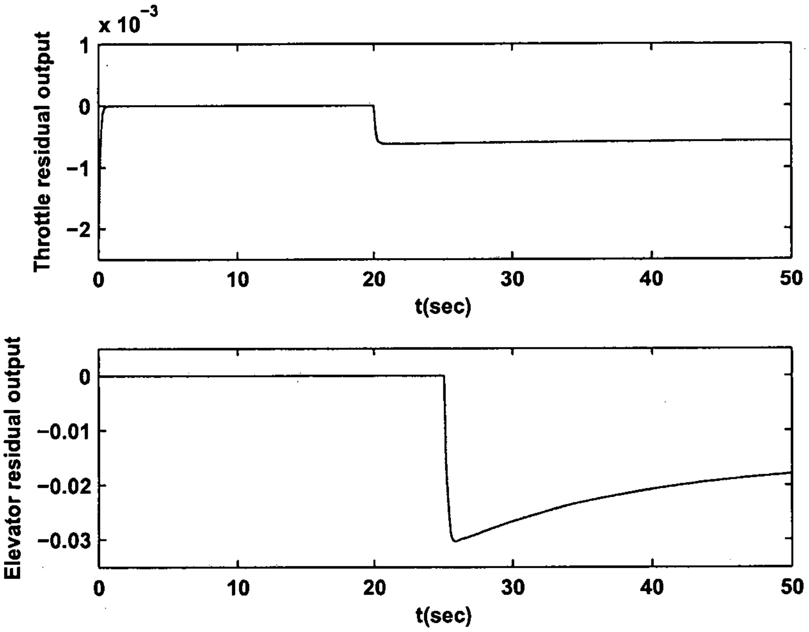


Figure 4.27: Detection observer residual outputs corresponding simultaneous faults in both input channel: Hard over fault in the throttle input at t=20 and Loss of performance (k=0.5) in the elevator input

4.3 ALTAV FDIR

In this chapter, the problem of actuator Fault Detection, Isolation and Recovery (FDIR) for Lighter-Than-Air Vehicles (ALTAV) is investigated. Fisrt, the FDI algorithms will be design using a nonlinear geometric approach. Due to the particular dynamics of ALTAV, six detection filters are designed for FDI of the four input channels of ALTAV. Four of the detection filters are designed such that each input channel affects only one of the actuators and the other two detection filters are affected by the corresponding two input channels. Using an appropriate combination of detection filters, one may detect and isolate the faults in all the input channels associated with single as well as simultaneous multiple actuators faults. Numerous simulation results demonstrate and show the excellent performance of the designed nonlinear detection filters.

Next, the safe mode operation of the ALTAV will be investigated to warranty the safety of the vehicle after occurrence of the fault. In this mode, one may just control the altitude of the vehicle while it is spinning around the z axis and the x and y coordinates of the ALTAV will converge to the constant value. Finally, the fault diagnosis algorithm has been proposed for identifying the type of fault in the safe mode operation. Figure 4.28 shows the flowchart of the FDIR procedure for the ALTAV system.

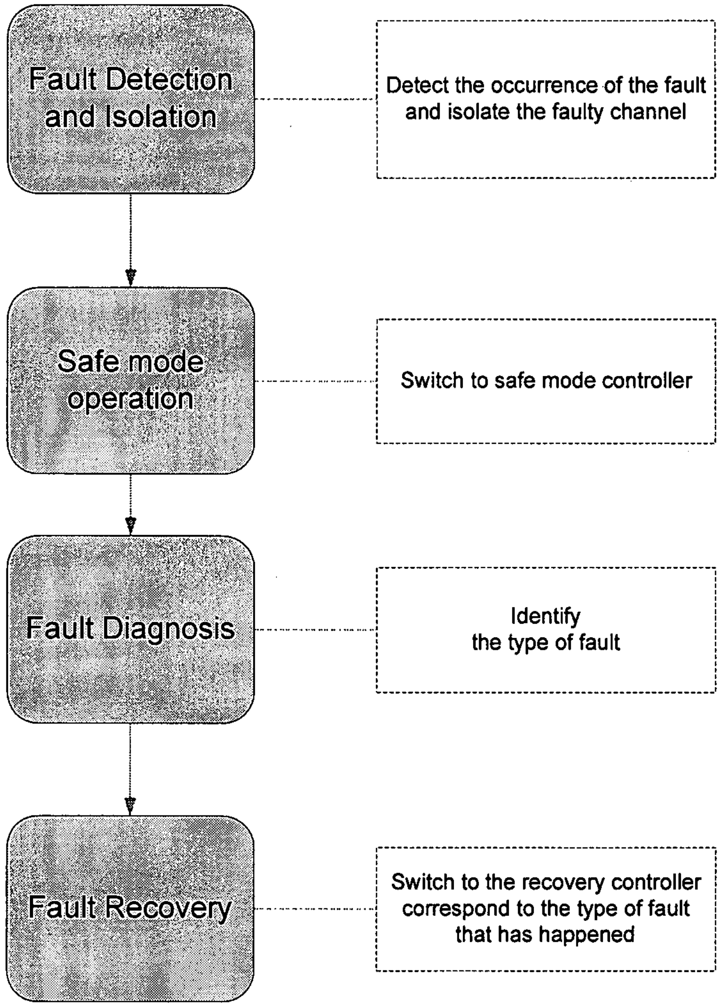

Fault Detection and Isolation
Detect the occurrence of the fault and isolate the faulty channel
Safe mode operation
Switch to safe mode controller
Fault Diagnosis
Identify the type of fault
Fault Recovery
Switch to the recovery controller correspond to the type of fault that has happened

Figure 4.28: Fault detection, isolation and recovery flowchart for the ALTAV system

4.3.1 Problem formulation

Lighter-Than-Air Vehicles (ALTAV) include balloons and airships. They use buoyancy to float in the air in ways that are similar to ships floating on the water. The Quanser ALTAV [?] provides a platform to demonstrate fault diagnosis methods for fleet of unmanned aerial vehicles. The Quanser ALTAV used in the simulations in this paper and in real world experiments, has six degrees of freedom vehicles, in which translational and rotational motions are described by six variables, and four control inputs are available corresponding to four motors on the vehicle.

The ALTAV system used in this paper and in real world experiments is six degrees of freedom unmanned aerial vehicle. The variables describing the motion are x, y, z, θ, γ and ϕ. These variables correspond to the translation in x, y and z directions and rotations about z, y and x axes (heading, pitch and roll) , respectively,. It should be noted that the system uses a "right handed" coordinate system with the positive z direction as down. The behavior of the ALTAV is governed by the following governing equations:

$$\begin{aligned}
M\ddot{x} &= \sum F_i sin(\gamma) - C_x\dot{x} \\
M\ddot{y} &= \sum F_i sin(\phi) - C_y\dot{y} \\
M\ddot{z} &= -\sum F_i cos(\gamma)cos(\phi) - F_B + Mg - C_z\dot{z} \\
J_\theta\ddot{\theta} &= (F_1 l - F_2 l + F_3 l - F_4 l)sin(\rho) - C_\theta\dot{\theta} \\
J_\gamma\ddot{\gamma} &= (F_1 l - F_3 l) - F_B L_B sin(\gamma) - C_\gamma\dot{\gamma} \\
J_\phi\ddot{\phi} &= -(F_2 l - F_4 l) - F_B L_B sin(\phi) - C_\phi\dot{\phi}
\end{aligned} \tag{4.35}$$

where M is the mass of the vehicle, $J_\phi, J_\gamma, J_\theta$ are the moments of inertia about x, y and z axes, respectively, F_B is the buoyant force resulting from the volume of helium in the vehicle, $F_i, i = 1, \cdots, 4$ are the force magnitude of motors, l is the perpendicular distance between the motors and vehicle center of gravity, C_i is the drag coefficient in the directions $i \in [x, y, z, \theta, \gamma, \phi]$ which serves as a damping term for the motion in that direction, and ρ is the angular offset from vertical of the motor thrust vectors.

Common actuator faults that are considered in this paper may include [?]: (i) freezing or lock in-place (LIP) fault, (ii) float fault, (iii) hard-over fault (HOF), and (iv) loss of effectiveness (LOE) fault. In the case of LIP fault, the actuator freezes at a particular situation and does not respond to subsequent commands. HOF is characterized by the actuator moving to its upper or lower position limits regardless of the commanded signal. The actuator speed of response is bounded by the actuator rate limits. Float fault occurs when the actuator floats with zero moment and does not contribute to the control authority. Loss of effectiveness is characterized by lowering the actuator gain with respect to its nominal value. Different types of actuator faults may be mathematically parameterized as follows:

$$u_{true} = \begin{cases} u_{cmd} & \text{No Failure Case} \\ k(t)u_{cmd} & 0 < \varepsilon \leq k(t) < 1, \forall t \geq t_F(\text{LOE}) \\ 0 & \forall t \geq t_F(\text{Float}) \\ u_{cmd}(t_F) & \forall t \geq t_F(\text{LIP}) \\ u_m \quad \text{or} \quad u_M & \forall t \geq t_F(\text{HOF}) \end{cases} \tag{4.36}$$

where t_F denotes the time instant of fault occurrence in the actuator, k denotes its effectiveness coefficient such that $k \in [\varepsilon, 1]$ and ε denotes its minimum effectiveness, and u_m and u_M denote the minimum and maximum value of the input respectively. The following general model

$$u_{true} = \sigma k u_{cmd} + (1-\sigma)\bar{u} \tag{4.37}$$

may integrate all the above cases into a single representation, where u_{true} is the actuator output, u_{cmd} is the output of the controller (which at the same time is an input to the actuator), $\sigma = 1$ and $k = 1$ in the no fault operating regime, $\sigma = 1$ and $\varepsilon < k < 1$ in the case of loss of effectiveness fault and $\sigma = 0$ in other types of fault scenarios. Finally $u_m \leq \bar{u} \leq u_M$ is the position at which the actuator locks in case of float, lock-in-place and hard-over faults. The objective of this section is to design the FDI scheme for the ALTAV system using the nonlinear geometric approach.

4.3.2 Nonlinear Geometric FDI Approach

In the following, the method presented in [?] is briefly reviewed. This scheme provides us with the necessary and sufficient conditions for solving the problem of detecting and isolating actuator faults for nonlinear systems. It is assumed that the nonlinear system may be described by the following nonlinear model:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + \sum_{i=1}^{m} g_i(x) m_i \\ y &= h(x) \end{aligned} \tag{4.38}$$

with state x is defined in a neighborhood $\mathcal{X}$ of the origin in $\mathbb{R}^n$, inputs $u \in \mathbb{R}^m$, $m_i \in \mathbb{R}^m, i = 1, \cdots, m$ and output $y \in \mathbb{R}^q$, in which $f(x)$, the m columns $g_1(x), \cdots, g_m(x)$ of $g(x)$, and $h(x)$ are nonlinear smooth mappings and $f(0) = 0, h(0) = 0$. The signals u and m_i's in (4.38) denote the input channels for control purposes and the actuator fault signals, respectively.

In nonlinear geometric FDI approach, one seeks at finding the residual signals $r_i(t), i = 1, \cdots, m$ such that each actuator fault m_i only affects the corresponding residual r_i and that does not affect other residuals. Each residual signal can be generated if one can find the largest observability codistribution Ω_i^* which is locally spanned by exact differentials and contained in $P_i^{\perp}$ and that satisfies the following necessary condition:

$$(span\{g_i\})^{\perp} + \Omega_i^* = T^*X \tag{4.39}$$

where $P_i = span\{g_1(x), \cdots, g_{i-1}(x), g_{i+1}(x), \cdots, g_m(x)\}$. In [?] an algorithm is proposed for finding such an observability codistribution.

If an observability codistribution Ω_i^* exists, one may then construct state and output diffeomorphisms such that the nonlinear system (4.38) can be represented in the following canonical form in the new coordinates(for details refer to [?]):

$$
\begin{aligned}
\dot{z}_1 &= f_1(z_1, z_2) + g_{1i}(z_1, z_2)u_i + g_{1i}(z_1, z_2)m_i \\
\dot{z}_2 &= f_2(z_1, z_2, z_3) + \sum_{j=1}^{m} g_{2j}(z_1, z_2, z_3)u_j + \sum_{j=1}^{m} g_{2j}(z_1, z_2, z_3)m_j \\
\dot{z}_3 &= f_3(z_1, z_2, z_3) + \sum_{j=1}^{m} g_{3j}(z_1, z_2, z_3)u_j + \sum_{j=1}^{m} g_{3j}(z_1, z_2, z_3)m_j \\
y_1 &= h_1(z_1) \\
y_2 &= z_2
\end{aligned} \tag{4.40}
$$

Corresponding to the new state space representation, only the fault signal m_i can affect the z_1-subsystem, and since z_2 can be identified with output y_2, one may then interpret $z_2$as an auxiliary and independent input to the z_1-subsystem. Due to the main property of the unobservability codistribution, the z_1-subsystem satisfies observability rank conditions and one can therefore design an asymptotic observer for the state z_1. Hence, the problem of detecting and isolating faults in the original nonlinear system (4.38) reduces to that of designing an observer for the observable z_1-subsystem.

4.3.3 FDI design for the ALTAV system

In this section, FDI residual generators are designed for the four input channels of the ALTAV system. The state space equations of the ALTAV system may be rewritten as follows:

$$
\begin{aligned}
\dot{X} &= f(X) + g_1(X)F_1 + g_2(X)F_2 + g_3(X)F_3 + g_4(X)F_4 \\
Y &= X + v
\end{aligned} \tag{4.41}
$$

where $X^T = [x \; y \; z \; \dot{x} \; \dot{y} \; \dot{z} \; \theta \; \gamma \; \phi \; \dot{\theta} \; \dot{\gamma} \; \dot{\phi}]$ and v is the measurement noise and

$$g_1(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{M}sin(X_8) \\ \frac{1}{M}sin(X_9) \\ -\frac{1}{M}cos(X_8)cos(X_9) \\ 0 \\ 0 \\ 0 \\ \frac{l}{J_\theta}sin(\rho) \\ \frac{l}{J_\gamma} \\ 0 \end{bmatrix} \quad g_2(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{M}sin(X_8) \\ \frac{1}{M}sin(X_9) \\ -\frac{1}{M}cos(X_8)cos(X_9) \\ 0 \\ 0 \\ 0 \\ -\frac{l}{J_\theta}sin(\rho) \\ 0 \\ -\frac{l}{J_\phi} \end{bmatrix}$$

$$g_3(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{M}sin(X_8) \\ \frac{1}{M}sin(X_9) \\ -\frac{1}{M}cos(X_8)cos(X_9) \\ 0 \\ 0 \\ 0 \\ \frac{l}{J_\theta}sin(\rho) \\ -\frac{l}{J_\gamma} \\ 0 \end{bmatrix} \quad g_4(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{M}sin(X_8) \\ \frac{1}{M}sin(X_9) \\ -\frac{1}{M}cos(X_8)cos(X_9) \\ 0 \\ 0 \\ 0 \\ -\frac{l}{J_\theta}sin(\rho) \\ 0 \\ \frac{l}{J_\phi} \end{bmatrix}$$

For detecting a fault in the input channel F_1, one should determine a new set of states that are affected by F_1 and decoupled from other channels F_2, F_3 and F_4. Towards this end, the largest unobservability codistribution Ω_1 which is contained in $P_1^\perp$ should be found where $P_1 = span\{g_2(X), g_3(X), g_4(X)\}$ and also the following condition should hold:

$$span\{g_1(X)\} \not\subset \Omega_1^\perp \tag{4.42}$$

Condition (4.42 is the necessary condition for decoupling the faults in nonlinear systems. In the ALTAV model; since full state measurements are available, then

$$\Omega_1 = P_1^\perp \tag{4.43}$$

According to the above unobservability distribution, the following state may be found which is affected by F_1 and decoupled from other input channels.

$$z_1 = 2J_\gamma sin(\rho)sin(X_8)X_{11} + J_\theta sin(X_8)X_{10} + lMsin(\rho)X_4$$

Repeating the same approach, one may find the following set of states such that for each $i = 2, 3, 4$, z_i is affected by F_i and decoupled from other input channels, namely

$$\begin{aligned} z_2 &= -2J_\phi sin(\rho)sin(X_8)X_{12} - J_\theta sin(X_8)X_{10} + lMsin(\rho)X_4 \\ z_3 &= -2J_\gamma sin(\rho)sin(X_8)X_{11} + J_\theta sin(X_8)X_{10} + lMsin(\rho)X_4 \\ z_4 &= 2J_\phi sin(\rho)sin(X_8)X_{12} - J_\theta sin(X_8)X_{10} + lMsin(\rho)X_4 \end{aligned}$$

The state space representation of the above states may be expressed as follows:

$$\begin{aligned}
\dot{z}_1 =& -2C_\gamma sin(\rho)sin(X_8)X_{11} - 2F_BL_Bsin(\rho)sin^2(X_8) \\
&+ 2J_\gamma cos(X_8)sin(\rho)X_{11}^2 - C_\theta X_{10}sin(X_8) \\
&+ J_\theta X_{10}X_{11}cos(X_8) - C_xX_4lsin(\rho) \\
&+ 4lsin(\rho)sin(X_8)F_1 \\
\dot{z}_2 =& 2C_\phi sin(\rho)sin(X_8)X_{12} + 2F_BL_Bsin(\rho)sin(X_8)sin(X_9) \\
&- 2J_\phi cos(X_8)sin(\rho)X_{11}X_{12} + C_\theta X_{10}sin(X_8) \\
&- J_\theta X_{10}X_{11}cos(X_8) - C_xX_4lsin(\rho) \\
&+ 4lsin(\rho)sin(X_8)F_2 \\
\dot{z}_3 =& 2C_\gamma sin(\rho)sin(X_8)X_{11} + 2F_BL_Bsin(\rho)sin^2(X_8)] \\
&- 2J_\gamma cos(X_8)sin(\rho)X_{11}^2 - C_\theta X_{10}sin(X_8) \\
&+ J_\theta X_{10}X_{11}cos(X_8) - C_xX_4lsin(\rho) \\
&+ 4lsin(\rho)sin(X_8)F_3 \\
\dot{z}_4 =& -2C_\phi sin(\rho)sin(X_8)X_{12} - 2F_BL_Bsin(\rho)sin(X_8)sin(X_9) \\
&+ 2J_\phi cos(X_8)sin(\rho)X_{11}X_{12} + C_\theta X_{10}sin(X_8) \\
&- J_\theta X_{10}X_{11}cos(X_8) - C_xX_4lsin(\rho) \\
&+ 4lsin(\rho)sin(X_8)F_4
\end{aligned}$$

In the above state equations for $z_i, i = 1, \cdots, 4$ the input signal $F_i, i = 1, \cdots, 4$ is multiplied by the term $sin(X_8)$ $(sin(\gamma))$. This will introduce a problem in the fault detection decision making process. When there is a fault in one of the input channels and γ oscillates around zero, the corresponding residual will also oscillate about zero. Thus, one may not be able to explicitly determine the presence of a fault. To remedy this problem, two other states are introduced and considered for the purpose of unambiguous fault detection and isolation strategy as follows:

$$\begin{aligned}
z_5 &= X_{11} \\
z_6 &= X_{12} \qquad (4.44)
\end{aligned}$$

where the state space representation corresponding to these two states are given by:

$$\dot{z}_5 = -\frac{C_\gamma}{J_\gamma}z_5 - \frac{F_BL_B}{J_\gamma}sin(X_8) + \frac{l}{J_\gamma}(F_1 - F_3) \qquad (4.45)$$

$$\dot{z}_6 = -\frac{C_\phi}{J_\phi}z_6 - \frac{F_BL_B}{J_\phi}sin(X_9) + \frac{l}{J_\phi}(F_4 - F_2) \qquad (4.46)$$

As seen in the above equations, z_5 is affected by two input channels F_1 and F_3 and is decoupled from the other two channels. Furthermore state z_6 is complementary to z_5 i.e., it is affected by input channels F_2 and F_4 and is decoupled from other channels. Moreover, the coefficients of these input channels are constant and are not affected by the states of the ALTAV system. Consequently, the corresponding residuals do not show any oscillation around zero due to the particular behavior of the states of the ALTAV system.

Our proposed detection filters or nonlinear observers may now be designed for the above states. Since in the ALTAV system, all states are measurable, the following observers can be designed for the above states, namely

$$\begin{aligned}
\dot{\hat{z}}_1 =& -2C_\gamma sin(\rho)sin(Y_8)Y_{11} - 2F_BL_Bsin(\rho)sin^2(Y_8)\\
&+ 2J_\gamma cos(Y_8)sin(\rho)Y_{11}^2 - C_\theta Y_{10}sin(Y_8)\\
&+ J_\theta Y_{10}Y_{11}cos(Y_8) - C_xY_4lsin(\rho)\\
&+ 4lsin(\rho)sin(Y_8)F_1 - k(z_1 - \hat{z}_1)\\
\dot{\hat{z}}_2 =& 2C_\phi sin(\rho)sin(Y_8)Y_{12} + 2F_BL_Bsin(\rho)sin(Y_8)sin(Y_9)\\
&- 2J_\phi cos(Y_8)sin(\rho)Y_{11}Y_{12} + C_\theta Y_{10}sin(Y_8)\\
&- J_\theta Y_{10}Y_{11}cos(Y_8) - C_xY_4lsin(\rho)\\
&+ 4lsin(\rho)sin(Y_8)F_2 - k(z_2 - \hat{z}_2)\\
\dot{\hat{z}}_3 =& 2C_\gamma sin(\rho)sin(Y_8)Y_{11} + 2F_BL_Bsin(\rho)sin^2(Y_8)]\\
&- 2J_\gamma 7cos(Y_8)sin(\rho)Y_{11}^2 - C_\theta Y_{10}sin(Y_8)\\
&+ J_\theta Y_{10}Y_{11}cos(Y_8) - C_xY_4lsin(\rho)\\
&+ 4lsin(\rho)sin(Y_8)F_3 - k(z_3 - \hat{z}_3)\\
\dot{\hat{z}}_4 =& -2C_\phi sin(\rho)sin(Y_8)Y_{12} - 2F_BL_Bsin(\rho)sin(Y_8)sin(Y_9)\\
&+ 2J_\phi cos(Y_8)sin(\rho)Y_{11}Y_{12} + C_\theta Y_{10}sin(Y_8)\\
&- J_\theta Y_{10}Y_{11}cos(Y_8) - C_xY_4lsin(\rho)\\
&+ 4lsin(\rho)sin(Y_8)F_4 - k(z_4 - \hat{z}_4)\\
\dot{\hat{z}}_5 =& -\frac{C_\gamma}{J_\gamma}\hat{z}_5 - \frac{F_BL_B}{J_\gamma}sin(Y_8) + \frac{l}{J_\gamma}(F_1 - F_3) - k(z_5 - \hat{z}_5)\\
\dot{\hat{z}}_6 =& -\frac{C_\phi}{J_\phi}\hat{z}_6 - \frac{F_BL_B}{J_\phi}sin(Y_9) + \frac{l}{J_\phi}(F_4 - F_2) - k(z_6 - \hat{z}_6)
\end{aligned}$$

where $k > 0$. Corresponding to above observers, the following residuals may then be generated.

$$\begin{aligned}
r_1 &= z_1 - \hat{z}_1\\
r_2 &= z_2 - \hat{z}_2\\
r_3 &= z_3 - \hat{z}_3\\
r_4 &= z_4 - \hat{z}_4\\
r_{13} &= z_{13} - \hat{z}_{13}\\
r_{24} &= z_{24} - \hat{z}_{24}
\end{aligned}$$

For each residual r_i, one can define the flag R_i such that whenever the corresponding residual exceeds its threshold, it will be set to 1 and when the residual is less than the threshold, it will be set to 0 and the initial value of all flags is 0. Then one can define the following detection flags $flag_1, flag_2, flag_3, flag_4, flag_{13}, flag_{24}$ that can be used to detect and isolate the faults in the input channels.

$$
\begin{aligned}
flag_1 &= R_1 \vee (R_{13} \wedge flag_1) \\
flag_2 &= R_2 \vee (R_{24} \wedge flag_2) \\
flag_3 &= R_3 \vee (R_{13} \wedge flag_3) \\
flag_4 &= R_4 \vee (R_{24} \wedge flag_4) \\
flag_{13} &= R_{13} \\
flag_{24} &= R_{24}
\end{aligned}
$$

Remark: According to the above detection flags, one can detect and isolate all faults in the vehicle except the scenarios that there are simultaneous faults in the input channels F_1 and F_3 such that one of the faults are intermittent. In this case both detection flags $flag_1$ and $flag_3$ remain 1. However this problem can be avoided by going to safe mode operation after detection of the fault and diagnosis the each input channel separately. The same problem can happen for the input channels F_2 and F_4.

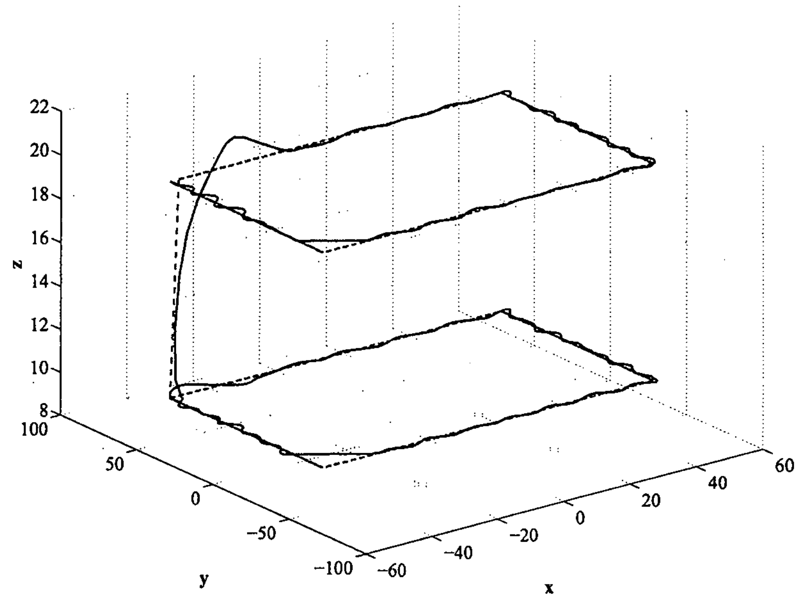


Figure 4.29: The desired reference trajectories (dashed line) and the actual ALTAV trajectories (solid line) in the healthy operation

4.3.4 Simulation Results

In this section, simulation results of our proposed nonlinear FDI scheme when applied to the nonlinear ALTAV model will be presented. Different actuator faults are considered in all the respective four input channels. We have considered two sources of disturbance on the ALTAV system as follows:

(i) Measurement noise v: Uniform random noise

(ii) Wind disturbance on x-axis: 0.12 N disturbance force on x direction

Figure 4.29 shows the desired reference trajectory and the actual ALTAV trajectory in the normal or healthy operation of the ALTAV. In this surveillance maneuver, the ALTAV first starts it motion from point (0,0,5), follows the square path in the x-y plane, changes its altitude to 10 and follows the same square in this altitude.

Figure 4.30 depicts the residuals of 6 observers when there is no fault in the ALTAV system. By considering the maximum value of residuals in the normal mode and some safety factor, the threshold value of 0.03 and 0.008 were considered for the residuals r_1, r_2, r_3, r_4 and r_{13}, r_{24} respectively.

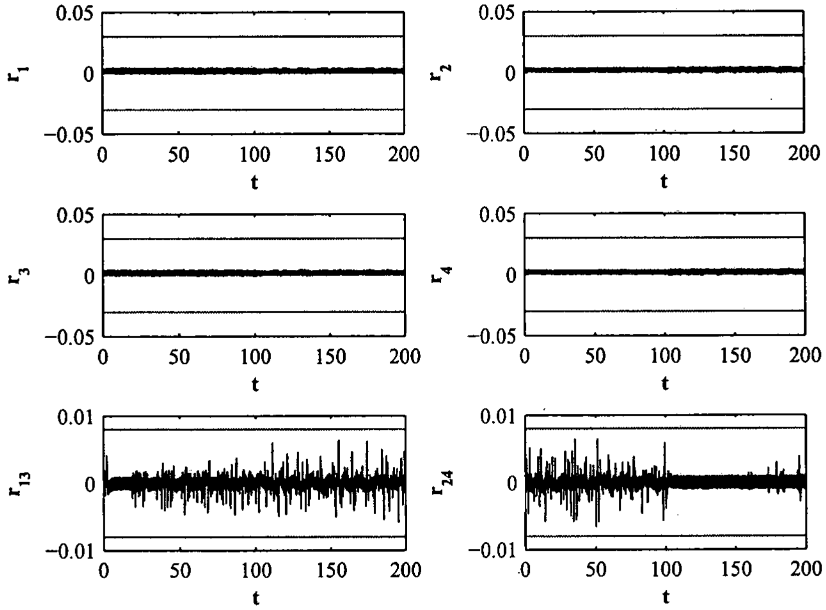


Figure 4.30: Residuals corresponding to healthy operation (No Fault)

The errors between the reference trajectory and the actual trajectory in each direction are considered to quantify the effect of fault on the vehicle motion. They are defined as

$$\begin{aligned} e_x &= \frac{x_{ref} - x}{x_r ef} \times 100 \\ e_y &= \frac{y_{ref} - y}{x_r ef} \times 100 \\ e_z &= \frac{z_{ref} - z}{x_r ef} \times 100 \end{aligned} \tag{4.47}$$

where $\{x, y, z\}_{ref}$ denote the reference trajectories of x, y and z directions, respectively. For the cases that the motion of the vehicle become unstable in specific direction, 100% is assigned to the corresponding error.

4.3.5 Single Fault scenarios

Figures 4.31, 4.32 and 4.33 show the residuals, the detection flags and the ALTAV states correspond to a float fault in F_1 actuator at $t = 100$ seconds, respectively. As it is shown in figure 4.31, the residuals r_1 and r_{13} exceed their thresholds but after about 30 seconds the residual r_1 starts to oscillate around zero due to oscillations present in γ. However by considering the residual r_{13} which remains above its threshold we may conclude that the fault still exists in the input channel F_1. According to 4.32 one can detect and isolate the float fault in F_1. Figure 4.34 show the actual trajectory of the ALTAV. Figures 4.35 and 4.36 depict the residuals and the detection flags correspond to a hard over fault in F_1 actuator at $t = 100$ seconds, respectively. In this faulty scenario, there is no oscillation in the residual r_1

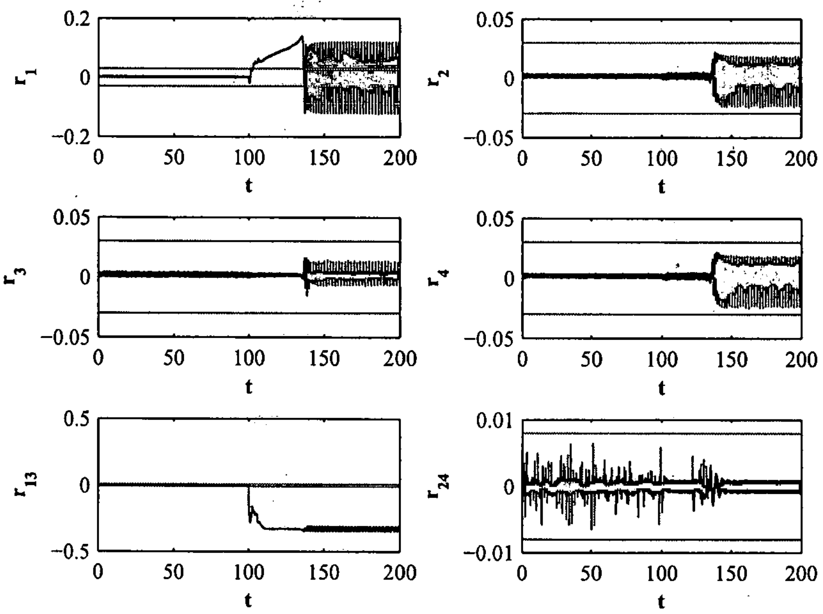


Figure 4.31: Residuals corresponding to a float fault in F_1 actuator

and one can easily detect and isolate this fault in F_1 actuator. Figure 4.37 show also that a hard over fault has a severe impact on the states of the ALTAV. Figures 4.38 and 4.39 depict the residuals and the detection flags correspond to a lock in place fault in F_1 actuator at $t = 100$ seconds, respectively. According to these figures, one can just detect but not isolate the fault in the input channels F_1 and F_3. Figure 4.40 show that this fault has a severe effect on the states of the ALTAV too. Figures 4.41 and 4.42 show the residuals and the detection flags correspond to a 20% loss of effectiveness in F_1 actuator at $t = 100$ seconds, respectively. According to these figures one can only detect the occurrence of the fault in one of the input channels F_1 and F_3 but cannot isolate between these two. Figures 4.44, 4.45 and 4.46 show the residuals, the detection flags and the ALTAV states correspond to a 50% loss of effectiveness in F_1 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_1 and F_3 but cannot isolate between these two. Figures 4.47, 4.48 and 4.49 show the residuals, the detection flags and the ALTAV states correspond to a 80% loss of effectiveness in F_1 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_1 and F_3 but cannot isolate between these two.

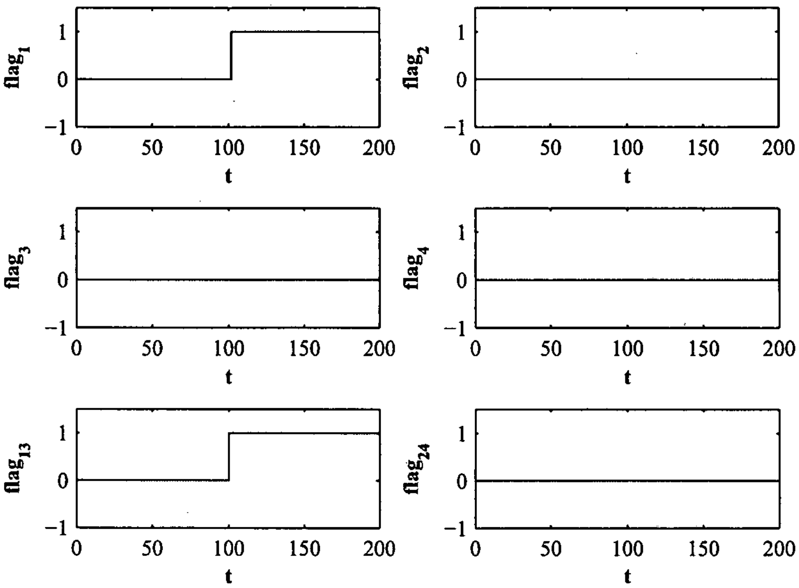


Figure 4.32: Detection flags corresponding to a float fault in F_1 actuator

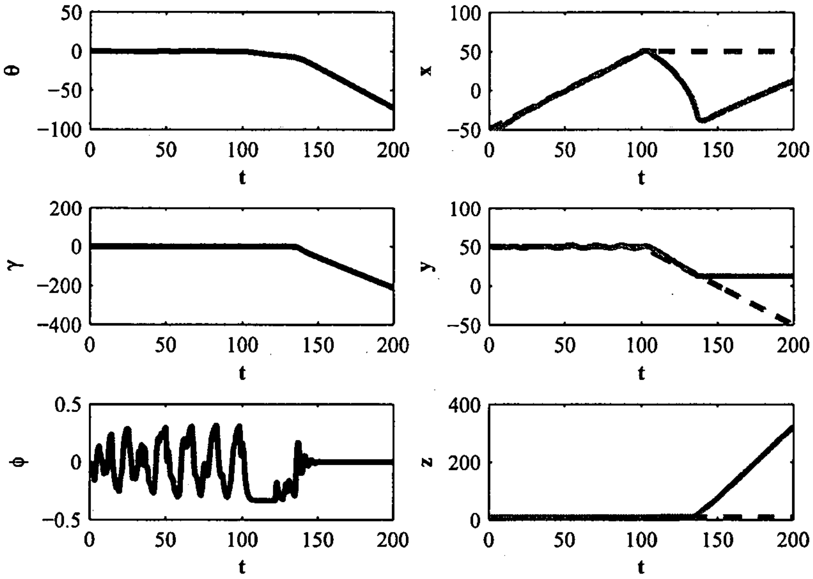


Figure 4.33: ALTAV states corresponding to a float fault in F_1 actuator

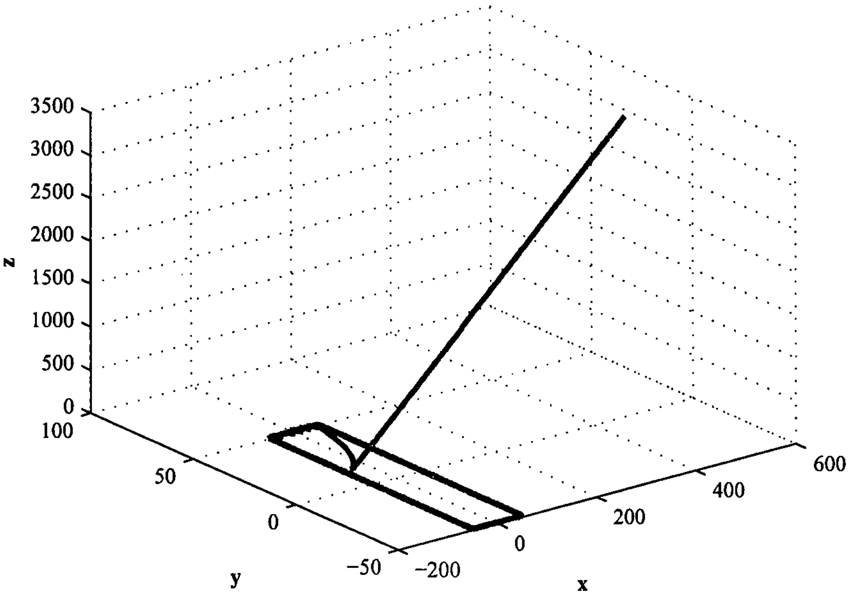


Figure 4.34: The desired reference trajectories (dashed line) and the actual ALTAV trajectories (solid line) corresponding to a float fault in F_1 actuator

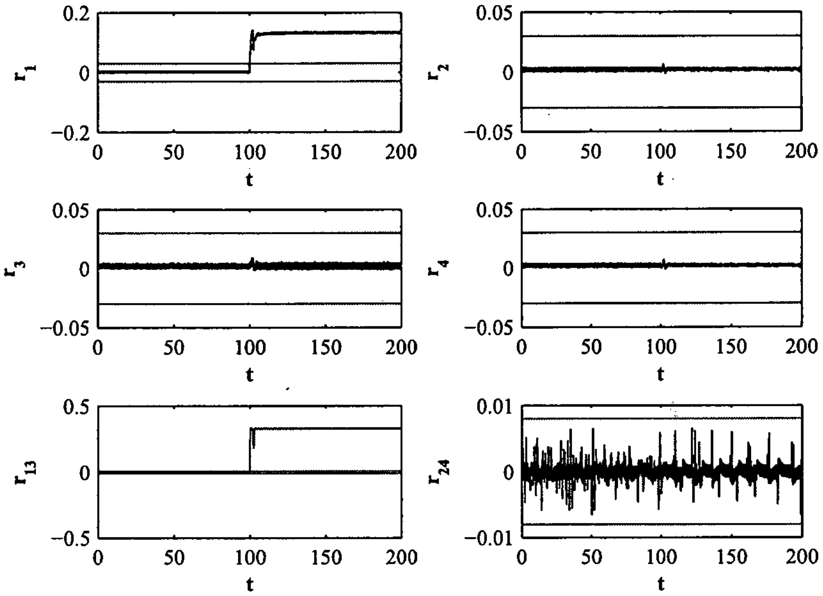


Figure 4.35: Residuals corresponding to a hard over fault in F_1 actuator

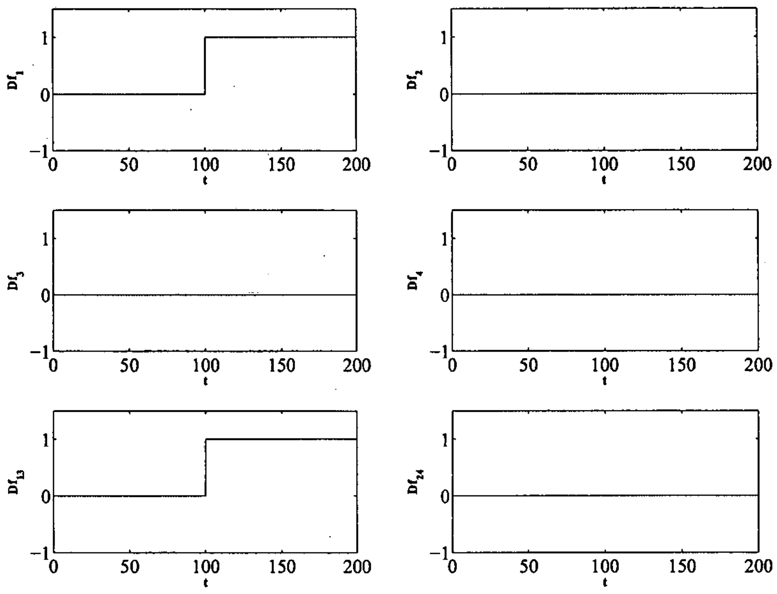


Figure 4.36: Detection flags corresponding to a hard over fault in F_1 actuator

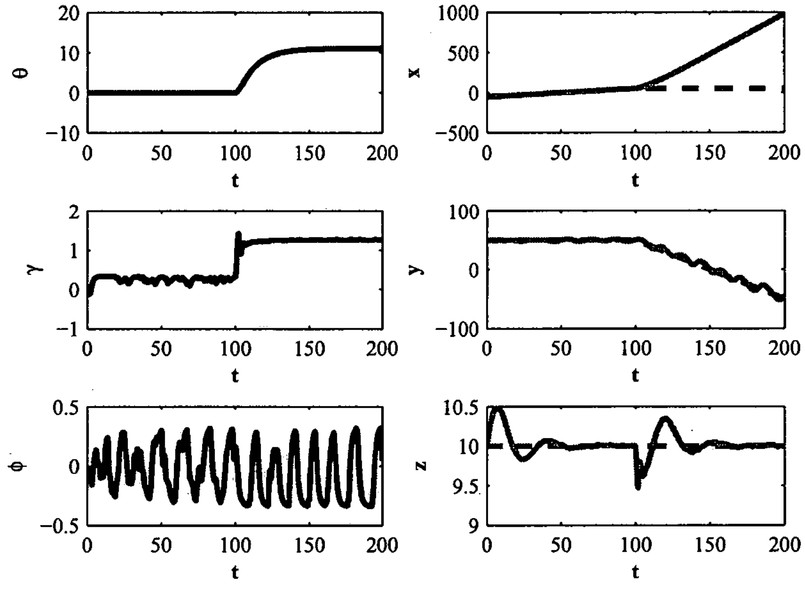


Figure 4.37: ALTAV states corresponding to a hard over fault in F_1 actuator

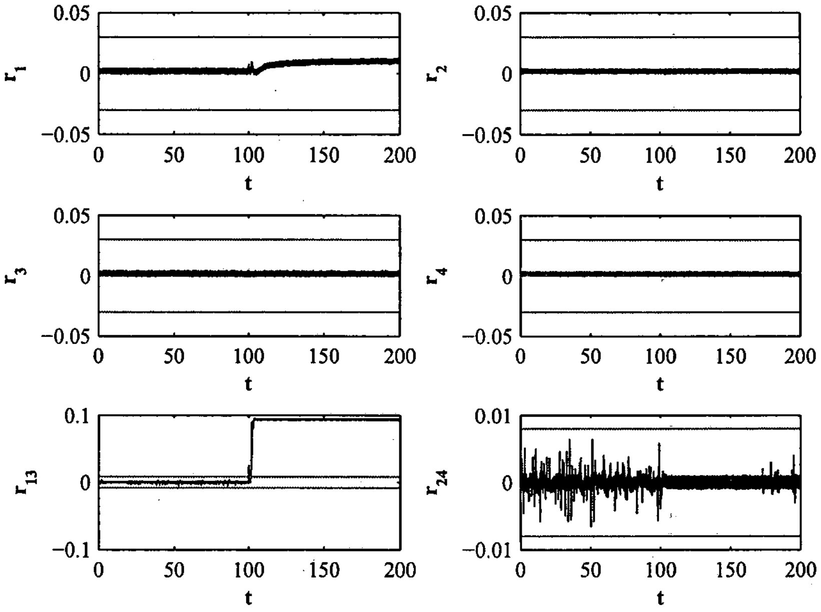


Figure 4.38: Residuals corresponding to a lock in place fault in F_1 actuator

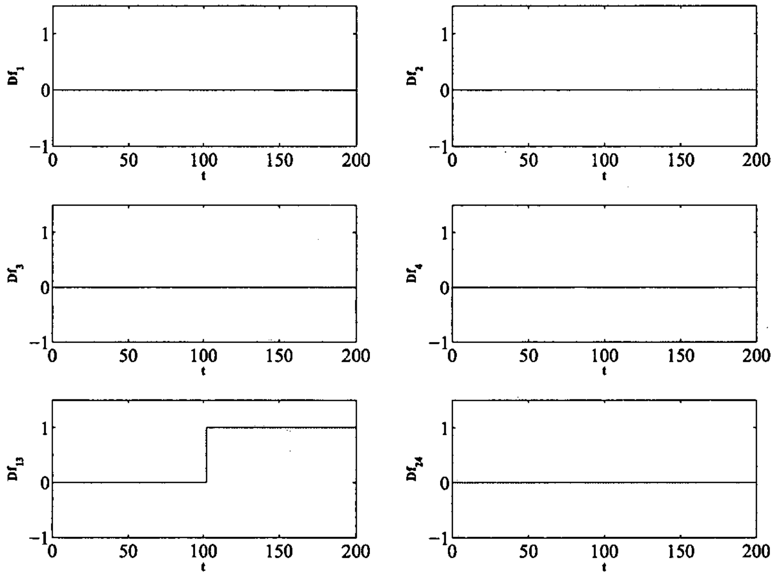


Figure 4.39: Detection flags corresponding to a lock in place fault in F_1 actuator

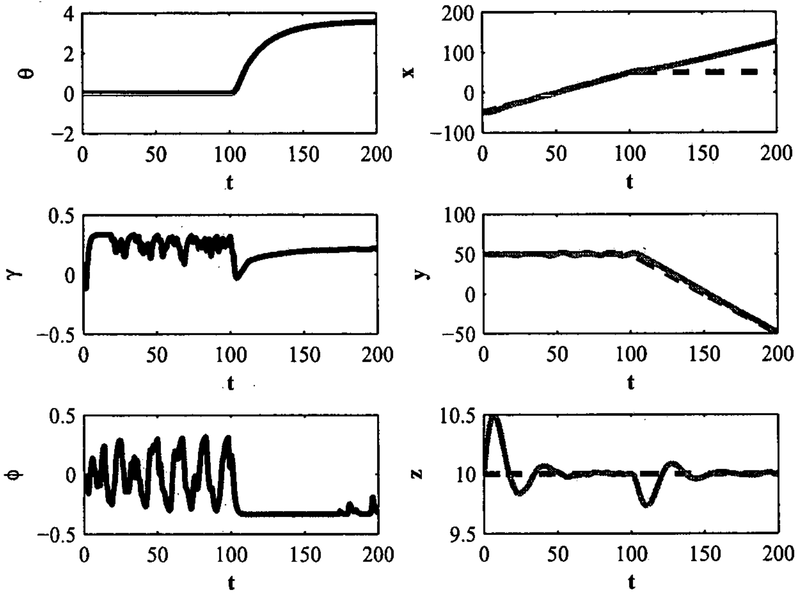


Figure 4.40: ALTAV states corresponding to a lock in place fault in F_1 actuator

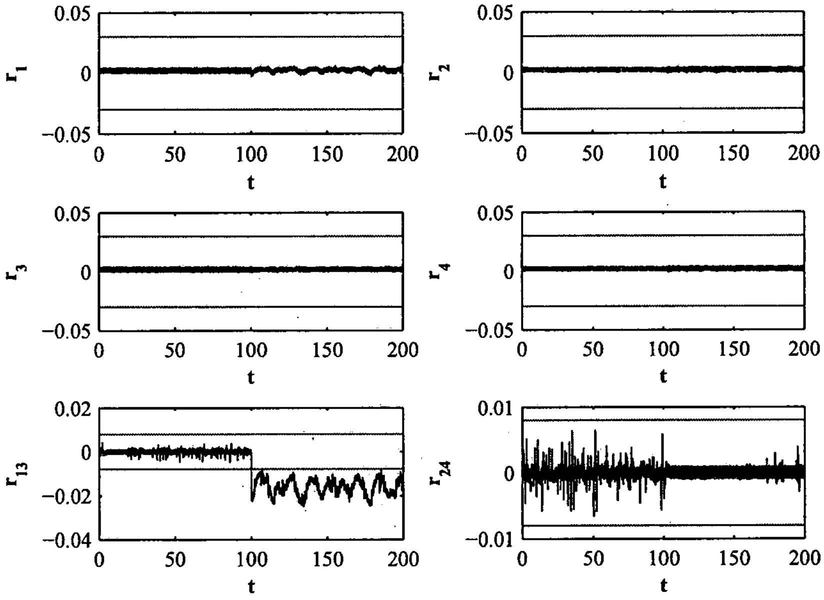


Figure 4.41: Residuals corresponding to a 20% loss of effectiveness fault in F_1 actuator

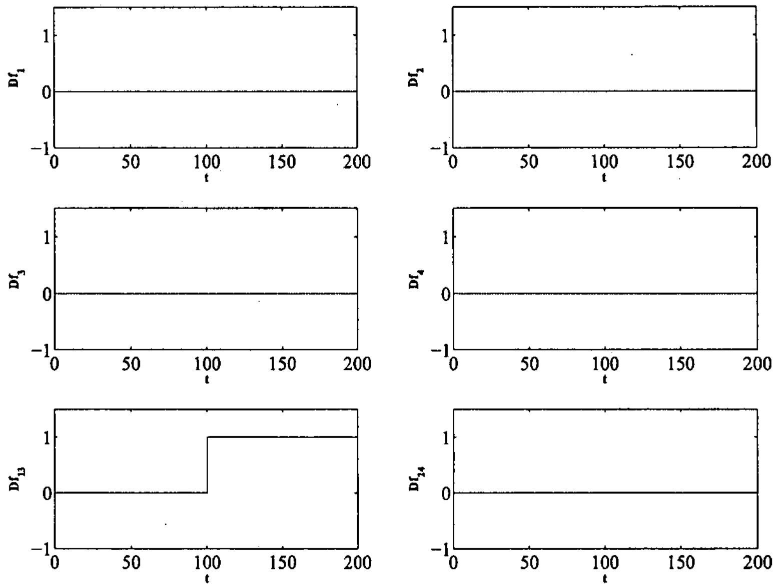


Figure 4.42: Detection flags corresponding to a 20% loss of effectiveness fault in F_1 actuator

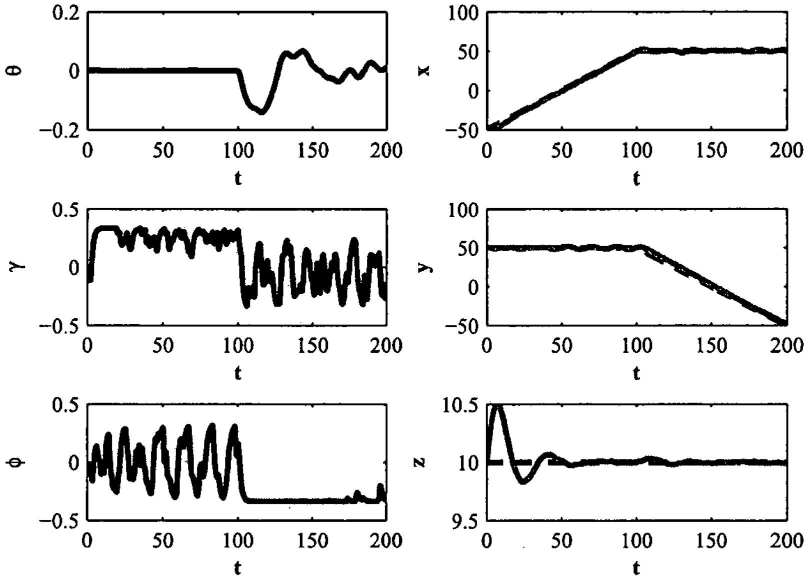


Figure 4.43: ALTAV states corresponding to a 20% loss of effectiveness fault in F_1 actuator

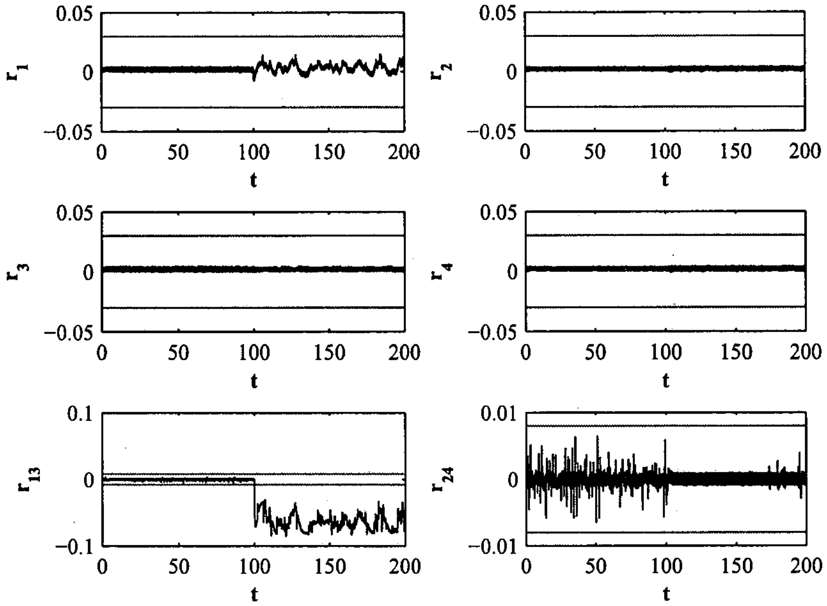


Figure 4.44: Residuals corresponding to a 50% loss of effectiveness fault in F_1 actuator

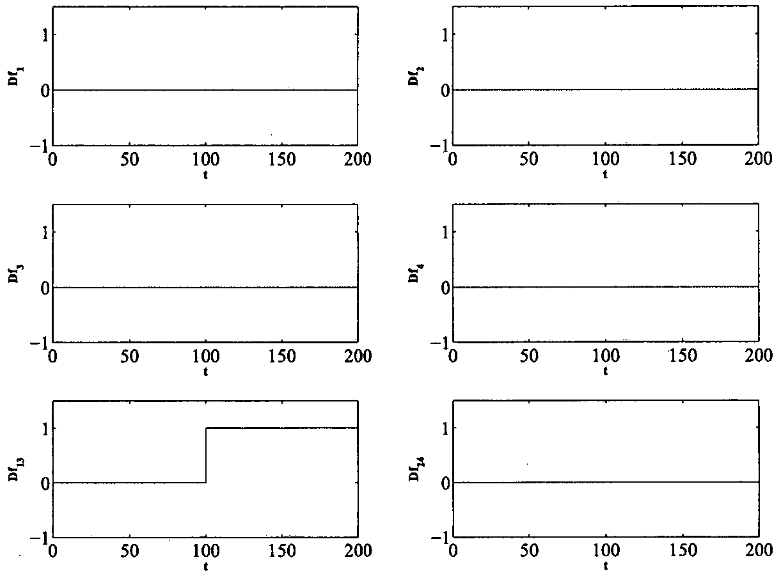


Figure 4.45: Detection flags corresponding to a 50% loss of effectiveness fault in F_1 actuator

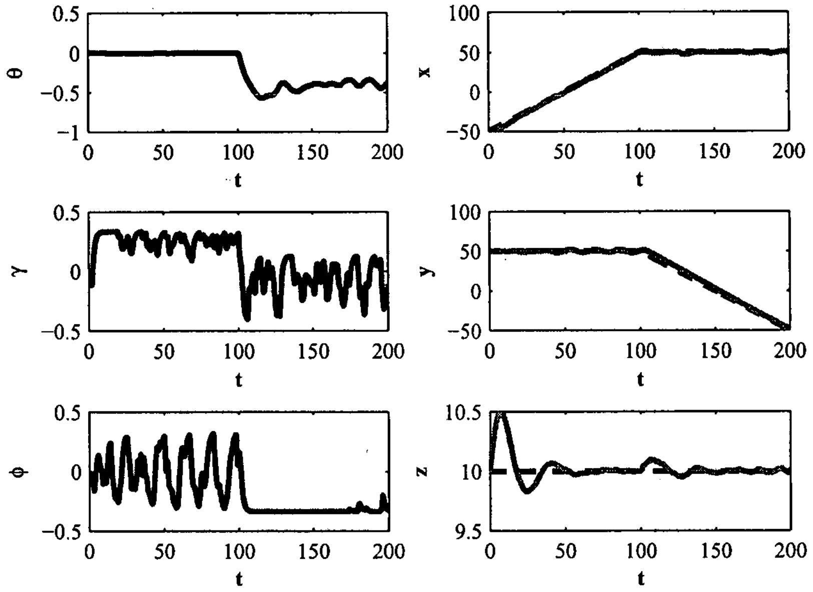


Figure 4.46: ALTAV states corresponding to a 50% loss of effectiveness fault in F_1 actuator

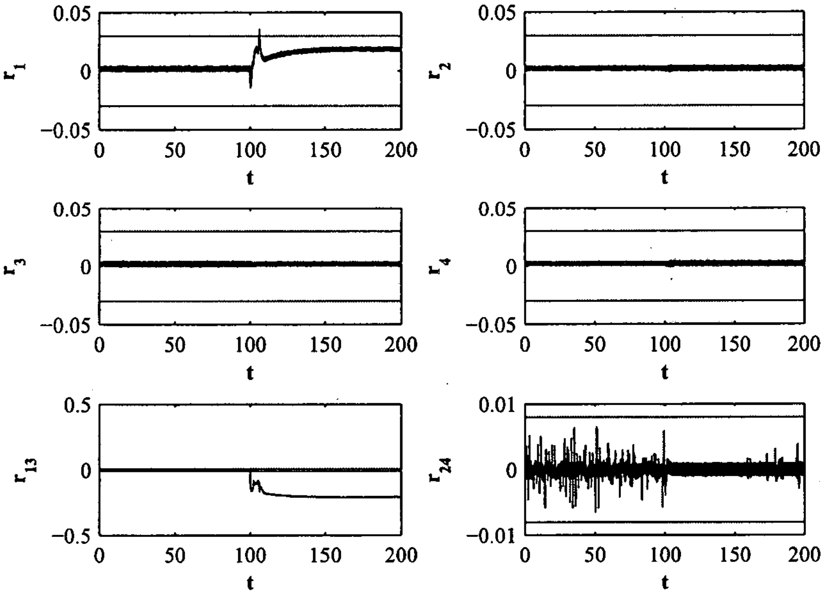


Figure 4.47: Residuals corresponding to a 80% loss of effectiveness fault in F_1 actuator

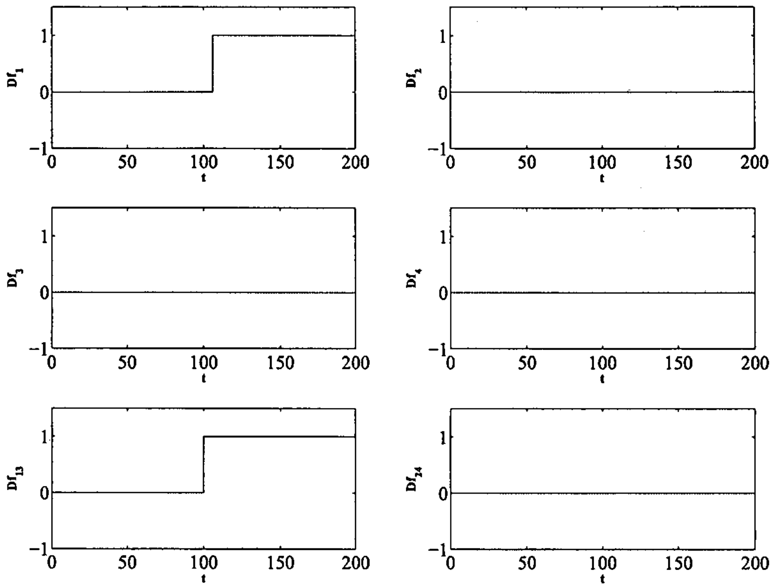


Figure 4.48: Detection flags corresponding to a 80% loss of effectiveness fault in F_1 actuator

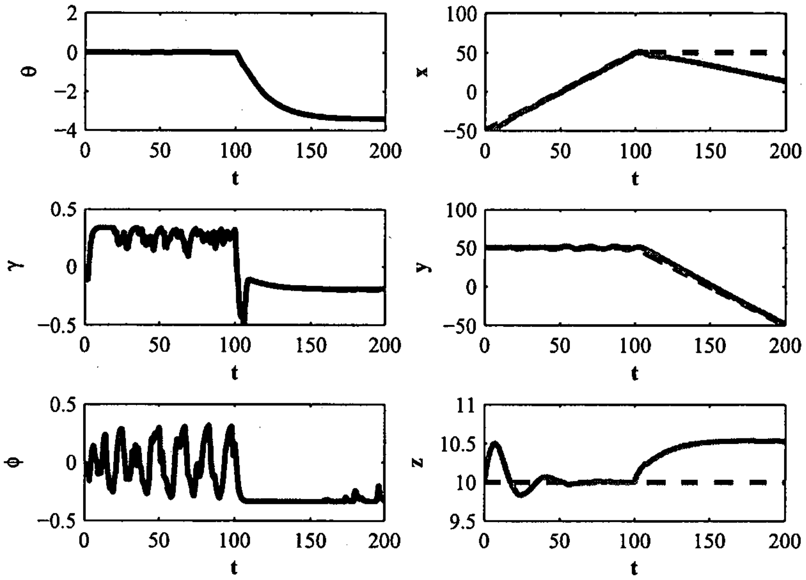


Figure 4.49: ALTAV states corresponding to a 80% loss of effectiveness fault in F_1 actuator

Type of fault	Detection	Isolation
Float	✓	✓
Hard over	✓	✓
Lock in place	✓	-
20% Loss of effectiveness	✓	-
50% Loss of effectiveness	✓	-
80% Loss of effectiveness	✓	✓

Table 4.9: Fault detection and isolation results for F_1 actuator

Table 4.9 summarizes the detection and isolation results for F_1 actuator.

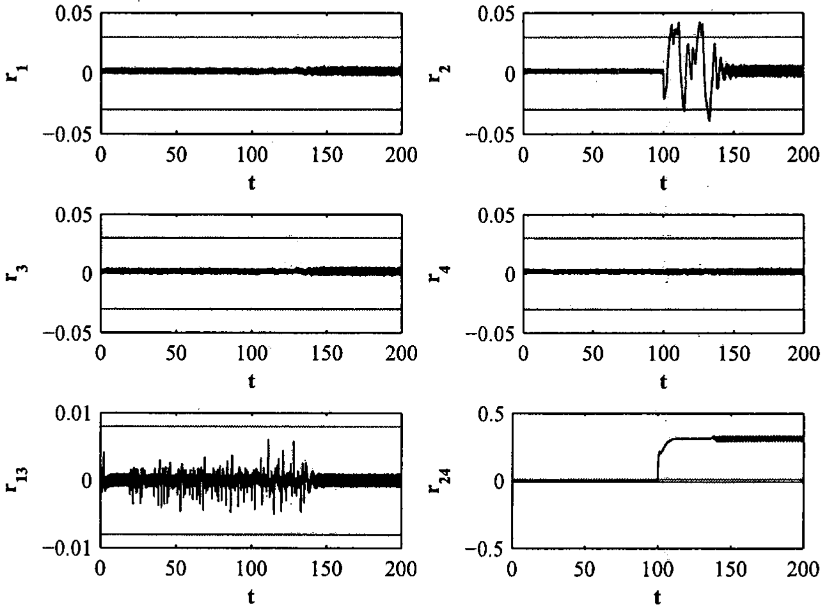


Figure 4.50: Residuals corresponding to a float fault in F_2 actuator

Figures 4.50, 4.51 and 4.33 show the residuals, the detection flags and the ALTAV states correspond to a float fault in F_2 actuator at $t = 100$ seconds, respectively. As it is shown in figure 4.50, the residuals r_2 and r_{24} exceed their thresholds but after about 30 seconds the residual r_2 starts to oscillate around zero due to oscillations present in γ. However by considering the residual r_{24} which remains above its threshold we may conclude that the fault still exists in the input channel F_2. According to 4.51 one can detect and isolate the float fault in F_2. Figures 4.53, 4.54 and 4.55 depict the residuals and the detection flags correspond to a hard over fault in F_2 actuator at $t = 100$ seconds, respectively. In this faulty scenario too, although there are some oscillations around zero in the residual r_2, one can still detect and isolate this fault in F_2 actuator. Figures 4.56, 4.57 and 4.58 depict the residuals, the detection flags and the ALTAV states correspond to a lock in place fault in F_2 actuator at $t = 100$ seconds, respectively. According to these figures, one can just detect but not isolate the fault in the input channels F_2 and F_4. Figures 4.59, 4.60 and 4.61 show the residuals, the detection flags and the ALTAV states correspond to a 20% loss of effectiveness in F_2 actuator at $t = 100$ seconds, respectively. According to these figures, one can just detect but not isolate the fault in the input channels F_2 and F_4. Figures 4.62, 4.63 and 4.64 show the residuals, the detection flags and the ALTAV states correspond to a 50% loss of effectiveness in F_2 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_2 and F_4 but cannot isolate between these two. Figure 4.64 shows the states of the ALTAV which demonstrates how this fault affect the ALTAV states. Figures 4.65, 4.66 and 4.67 show the residuals, the detection flags and ALTAV states correspond to a 80% loss of effectiveness in F_2 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_2 and F_4 but cannot isolate between these two.

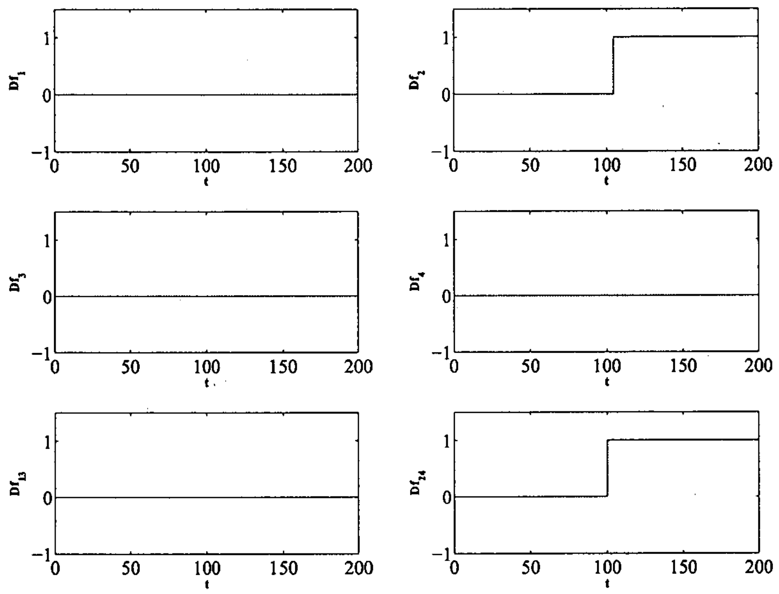


Figure 4.51: Detection flags corresponding to a float fault in F_2 actuator

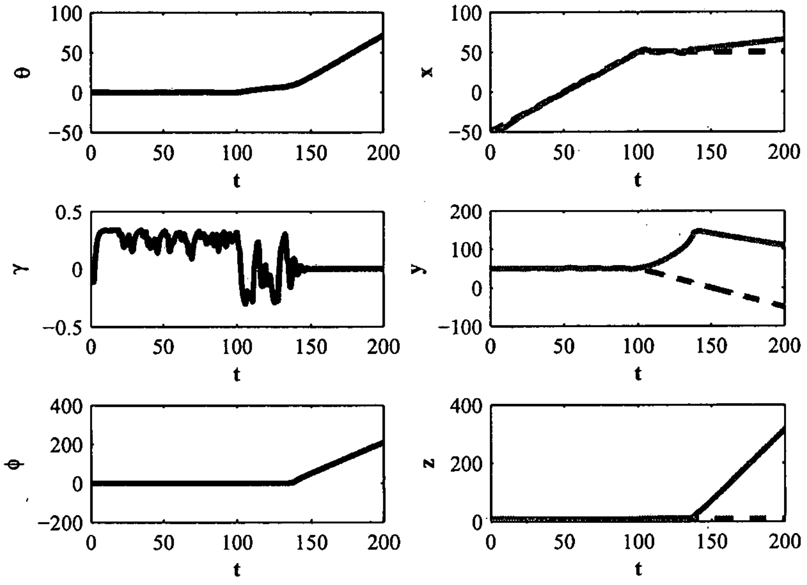


Figure 4.52: ALTAV states corresponding to a float fault in F_2 actuator

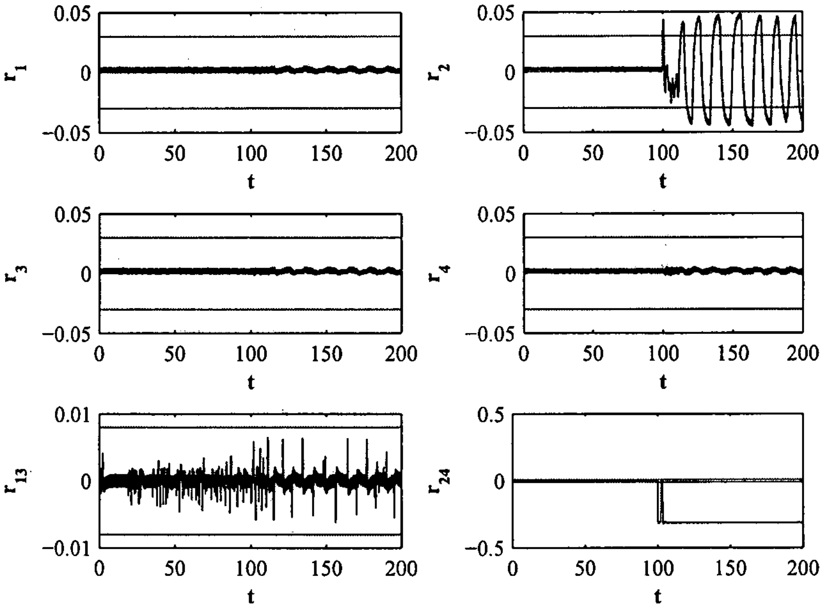


Figure 4.53: Residuals corresponding to a hard over fault in F_2 actuator

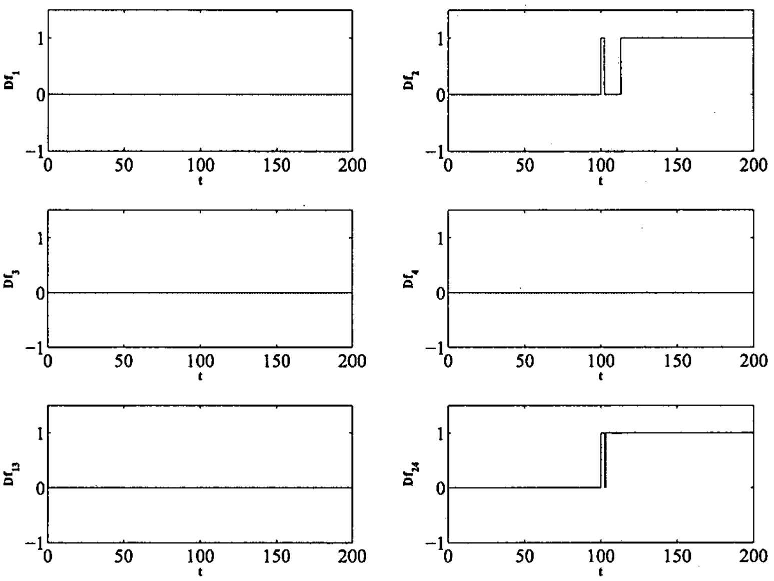


Figure 4.54: Detection flags corresponding to a hard over fault in F_2 actuator

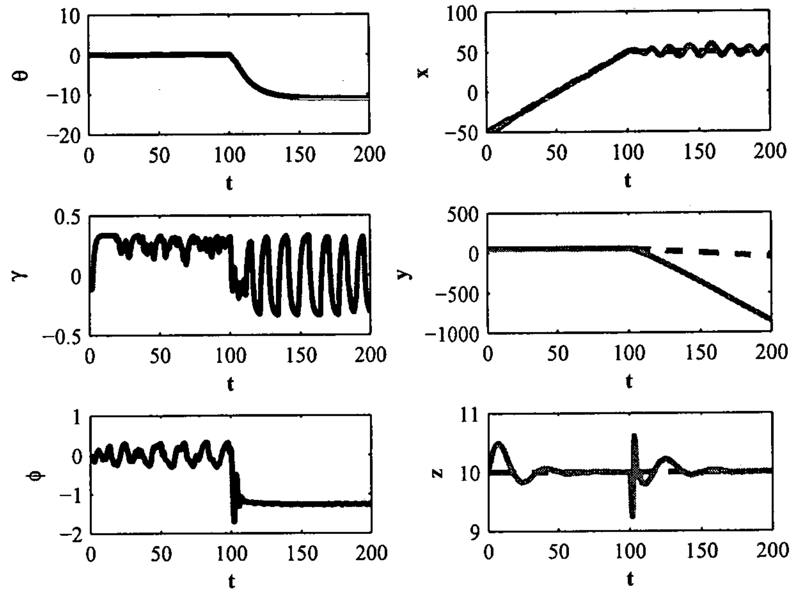


Figure 4.55: ALTAV states corresponding to a hard over fault in F_2 actuator

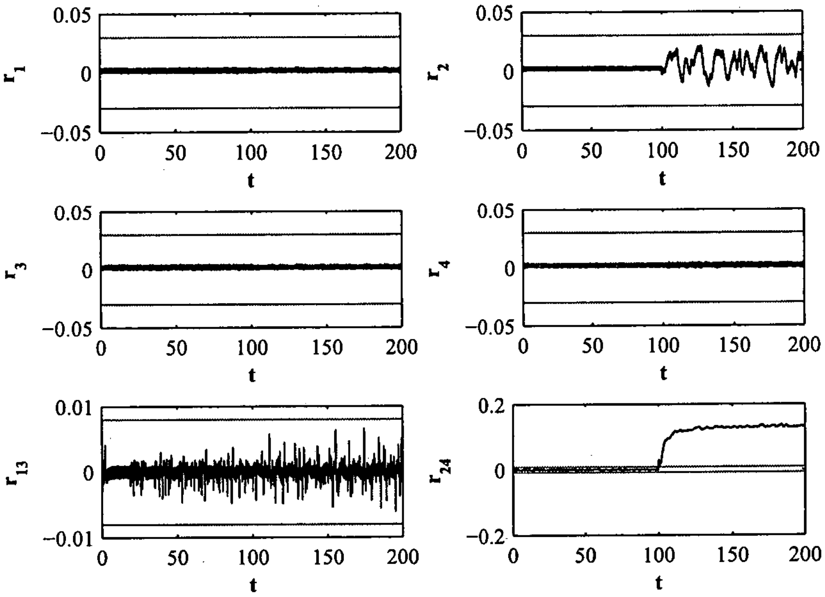


Figure 4.56: Residuals corresponding to a lock in place fault in F_2 actuator

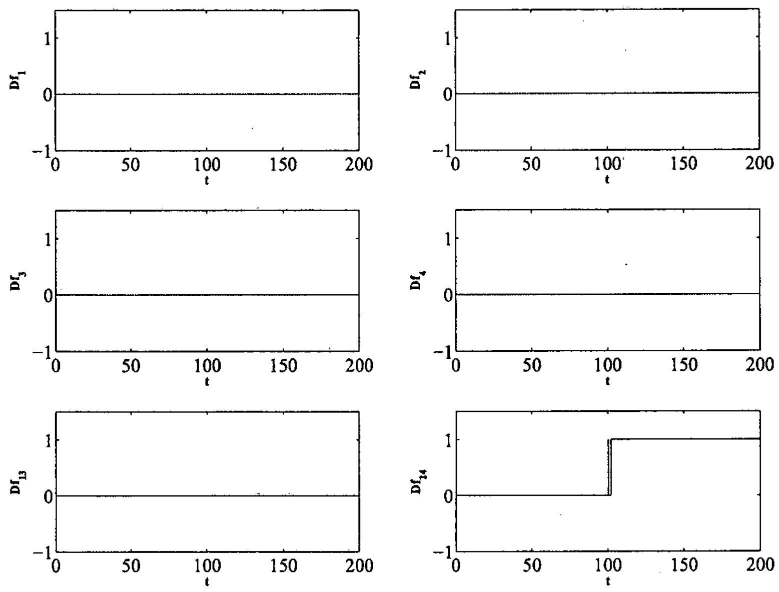


Figure 4.57: Detection flags corresponding to a lock in place fault in F_2 actuator

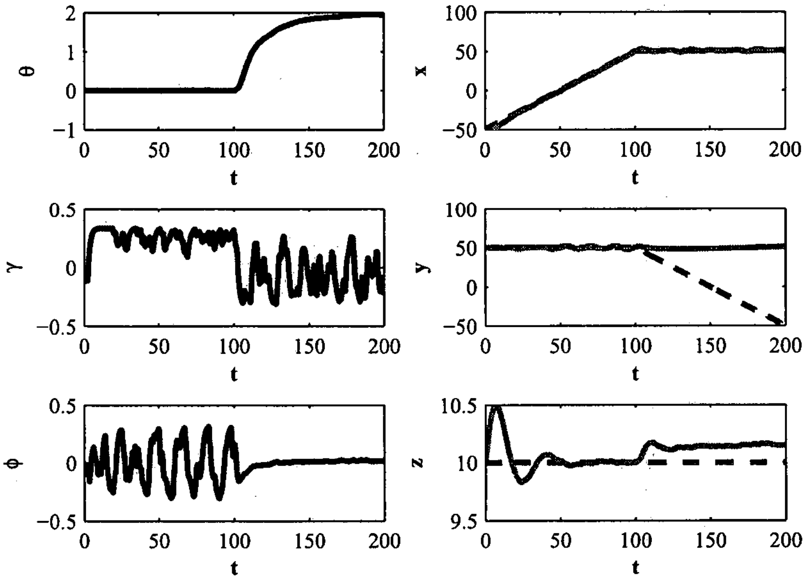


Figure 4.58: ALTAV states corresponding to a lock in place fault in F_2 actuator

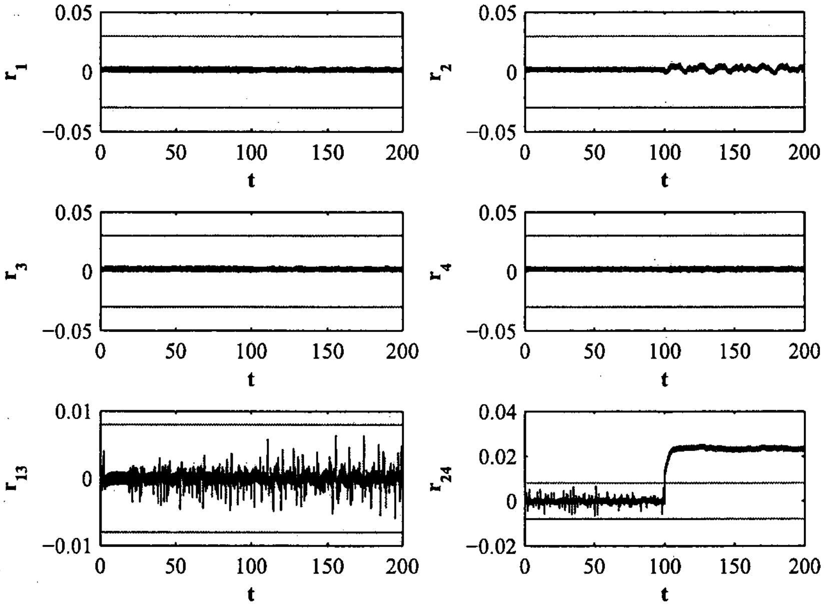


Figure 4.59: Residuals corresponding to a 20% loss of effectiveness fault in F_2 actuator

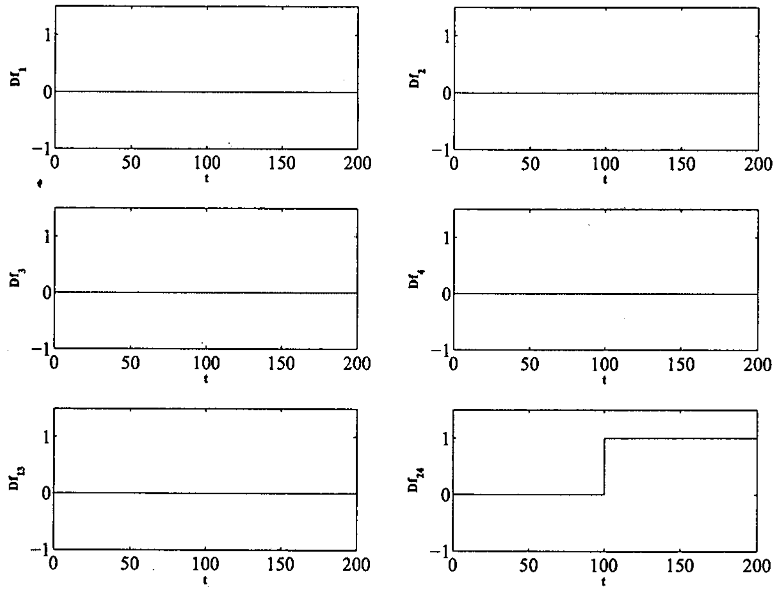


Figure 4.60: Detection flags corresponding to a 20% loss of effectiveness fault in F_2 actuator

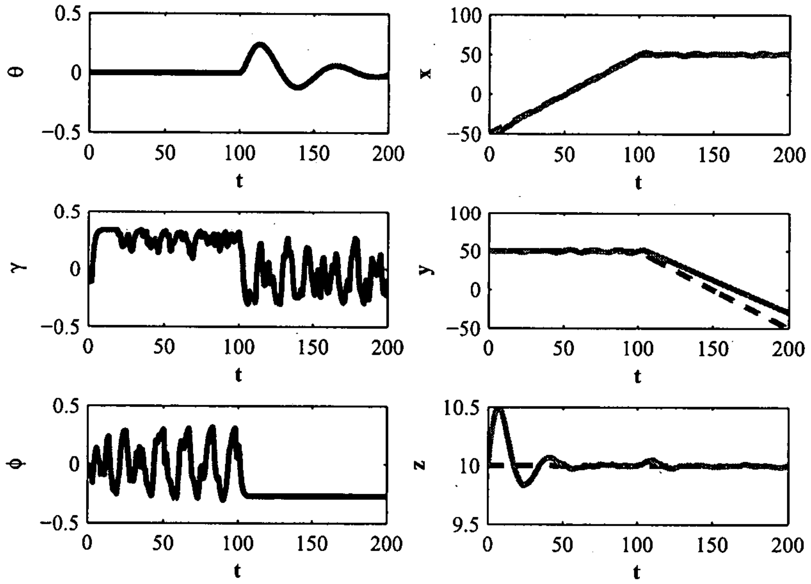


Figure 4.61: ALTAV states corresponding to a 20% loss of effectiveness fault in F_2 actuator

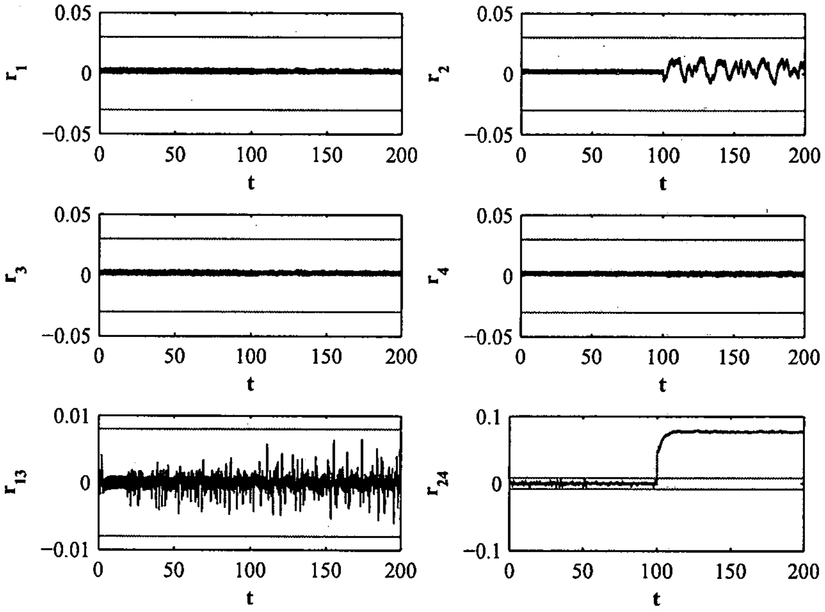


Figure 4.62: Residuals corresponding to a 50% loss of effectiveness fault in F_2 actuator

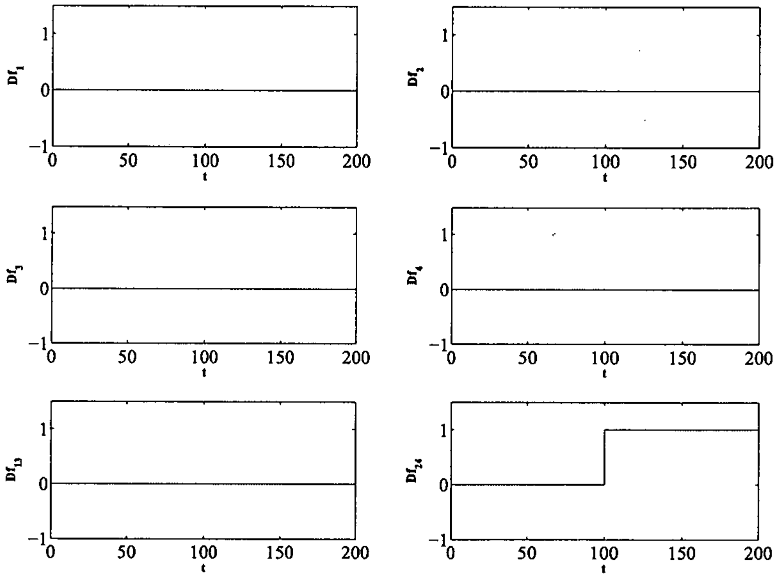


Figure 4.63: Detection flags corresponding to a 50% loss of effectiveness fault in F_2 actuator

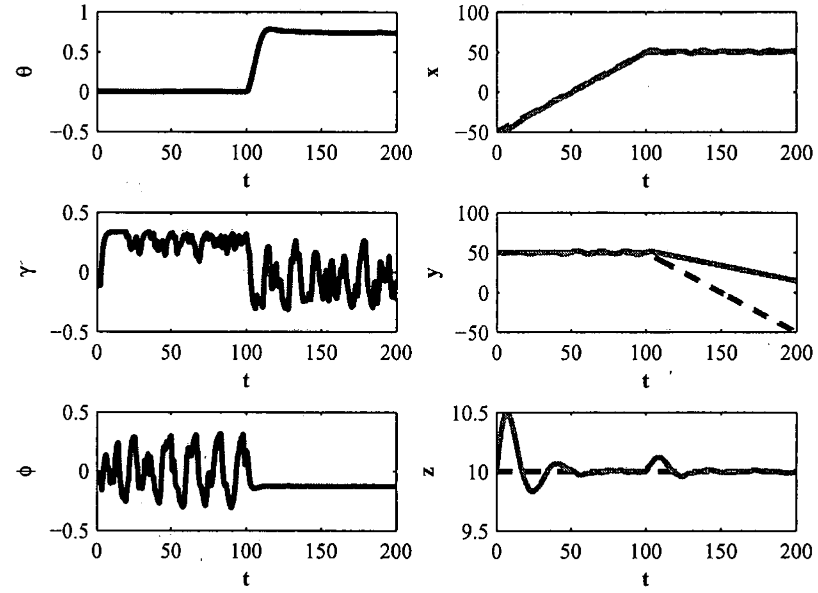


Figure 4.64: ALTAV states corresponding to a 50% loss of effectiveness fault in F_2 actuator

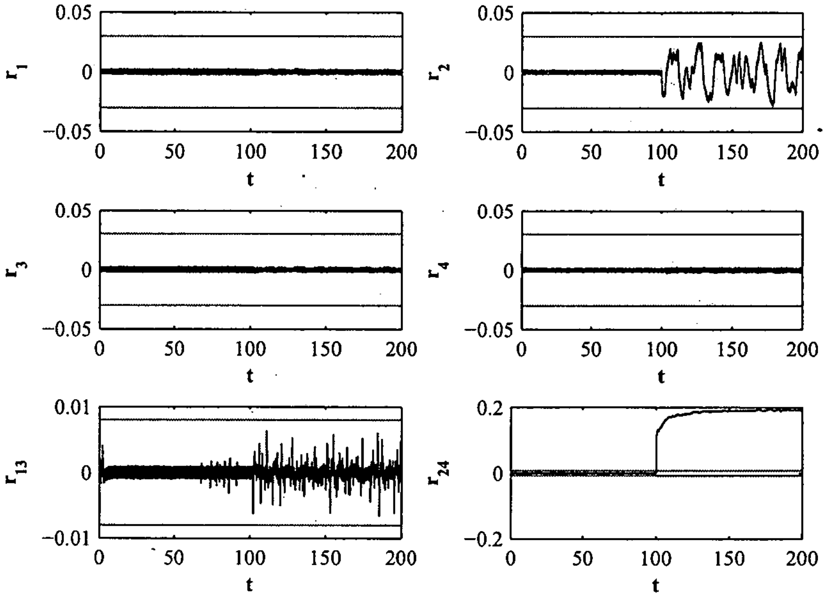


Figure 4.65: Residuals corresponding to a 80% loss of effectiveness fault in F_2 actuator

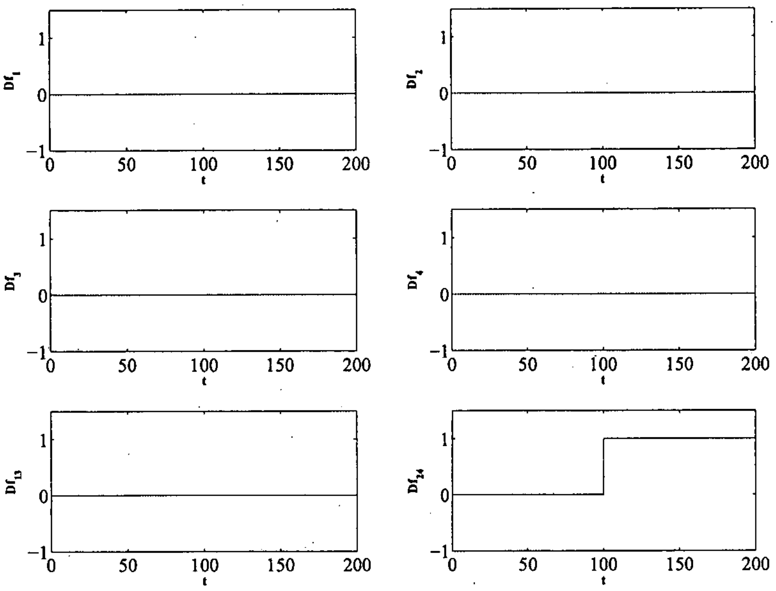


Figure 4.66: Detection flags corresponding to a 80% loss of effectiveness fault in F_2 actuator

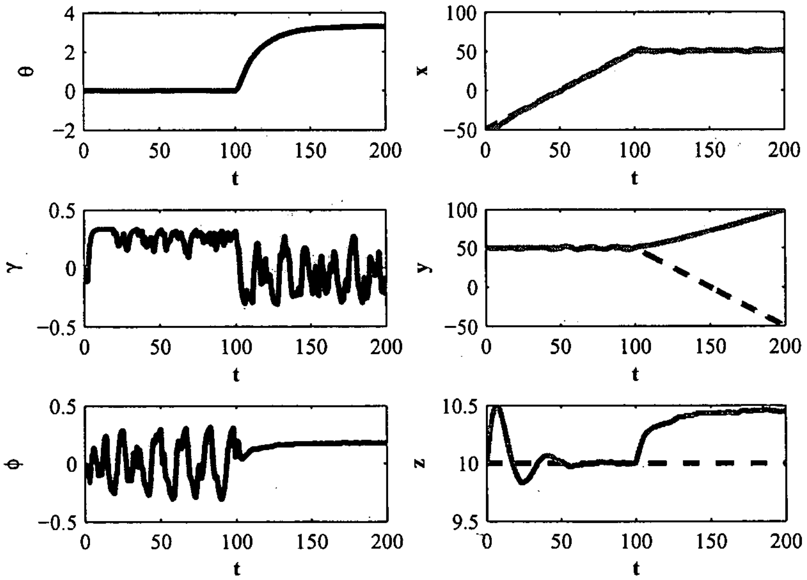


Figure 4.67: ALTAV states corresponding to a 80% loss of effectiveness fault in F_2 actuator

Type of fault	Detection	Isolation
Float	✓	✓
Hard over	✓	✓
Lock in place	✓	-
20% Loss of effectiveness	✓	-
50% Loss of effectiveness	✓	-
80% Loss of effectiveness	✓	-

Table 4.10: Fault detection and isolation results for F_2 actuator

Table 4.10 summarized the detection and isolation results for F_2 actuator and table ?? shows the effect of fault on tracking of reference trajectories.

Type of fault	Detection	Isolation
Float	✓	✓
Hard over	✓	✓
Lock in place	✓	-
20% Loss of effectiveness	✓	-
50% Loss of effectiveness	✓	-
80% Loss of effectiveness	✓	-

Table 4.11: Fault detection and isolation results for F_3 actuator

Figures 4.68, 4.69 and 4.68 show the residuals, the detection flags and the ALTAV states correspond to a float fault in F_3 actuator at $t = 100$ seconds, respectively. As it is shown in figure 4.68, the residuals r_1 and r_{13} exceed their thresholds but after about 30 seconds the residual r_1 starts to oscillate around zero due to oscillations present in γ. However by considering the residual r_{13} which remains above its threshold we may conclude that the fault still exists in the input channel F_3. According to 4.69 one can detect and isolate the float fault in F_3. Figures 4.71, 4.72 and 4.73 depict the residuals, the detection flags and the ALTAV states correspond to a hard over fault in F_3 actuator at $t = 100$ seconds, respectively. In this faulty scenario, there is no oscillation in the residual r_1 and one can easily detect and isolate this fault in F_3 actuator. Figures 4.74, 4.75 and 4.76 depict the residuals, the detection flags and the ALTAV states correspond to a lock in place fault in F_3 actuator at $t = 100$ seconds, respectively. According to these figures, one can just detect but not isolate the fault in the input channels F_1 and F_3. Figures 4.77, 4.78 and 4.79 show the residuals and the detection flags correspond to a 20% loss of effectiveness in F_3 actuator at $t = 100$ seconds, respectively. According to these figures one can only detect the occurrence of the fault in the input channels F_1 and F_3. Figures 4.80, 4.81 and 4.82 show the residuals , the detection flags and the ALTAV states correspond to a 50% loss of effectiveness in F_3 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_1 and F_3 but cannot isolate between these two. Figures 4.83, 4.84 and 4.85 show the residuals, the detection flags and ALTAV states correspond to a 80% loss of effectiveness in F_3 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_1 and F_3 but cannot isolate between these two.

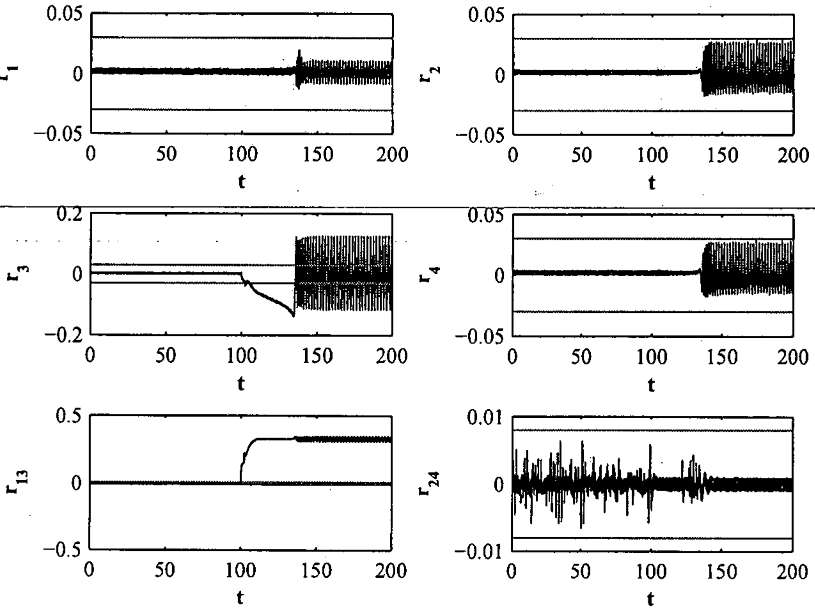


Figure 4.68: Residuals corresponding to a float fault in F_3 actuator

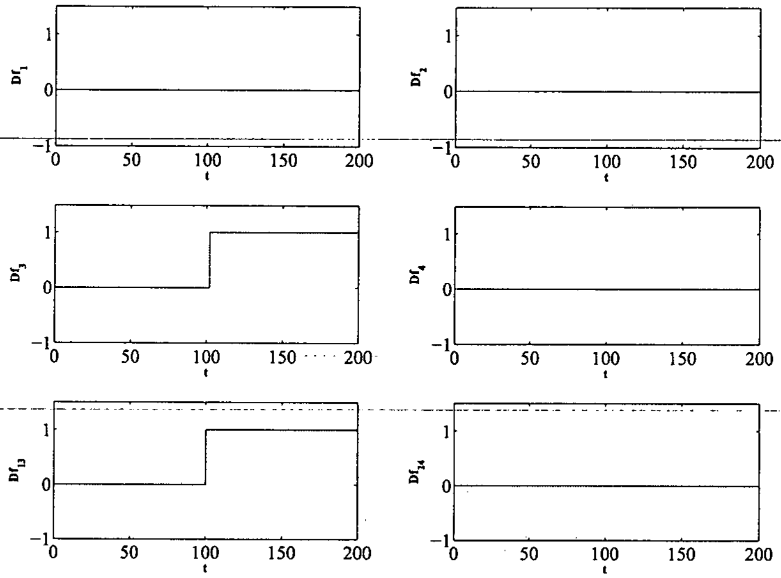


Figure 4.69: Detection flags corresponding to a float fault in F_3 actuator

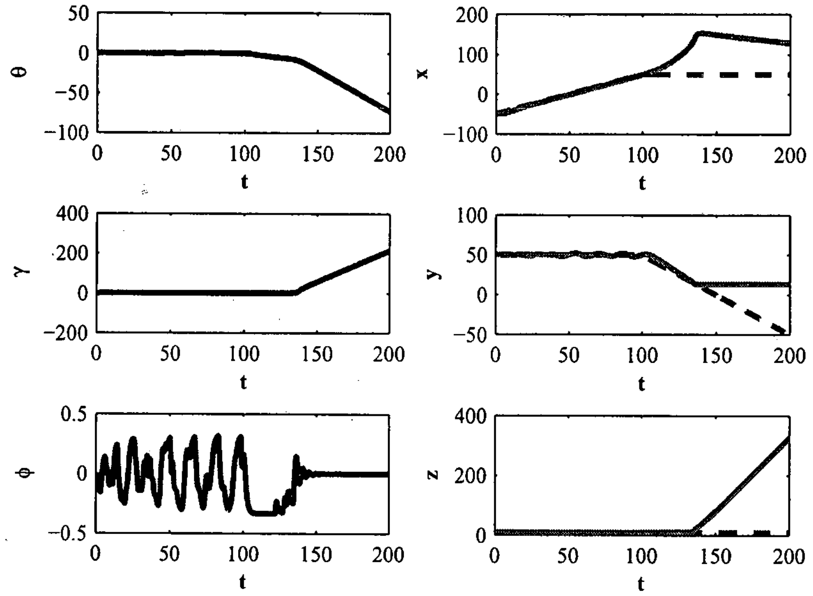


Figure 4.70: ALTAV states corresponding to a float fault in F_3 actuator

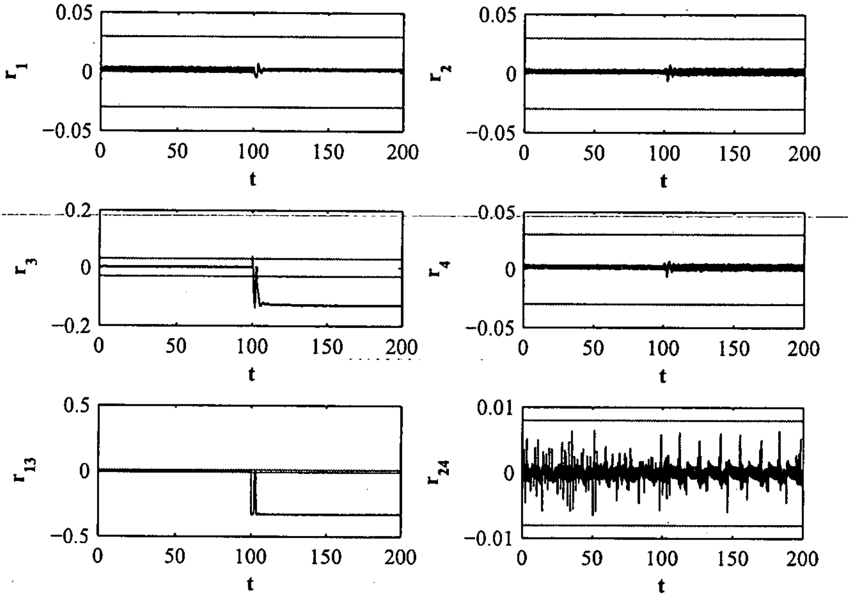


Figure 4.71: Residuals corresponding to a hard over fault in F_3 actuator

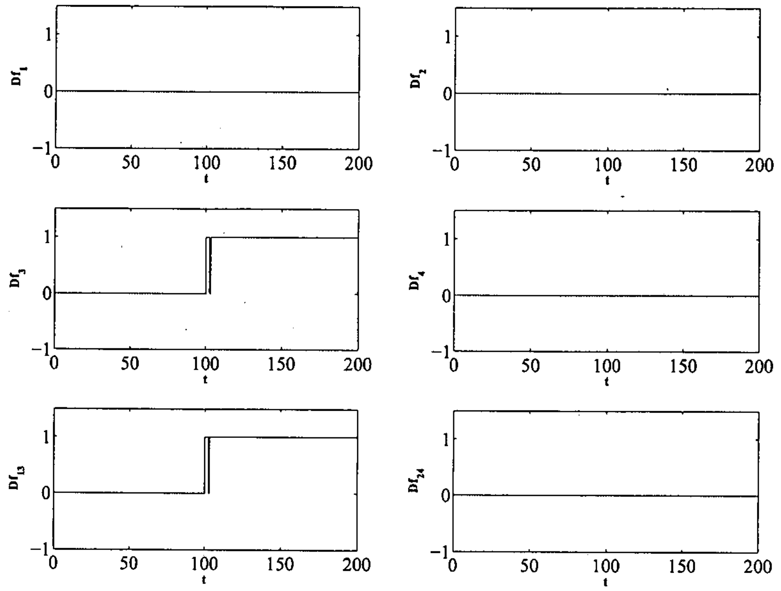


Figure 4.72: Detection flags corresponding to a hard over fault in F_3 actuator

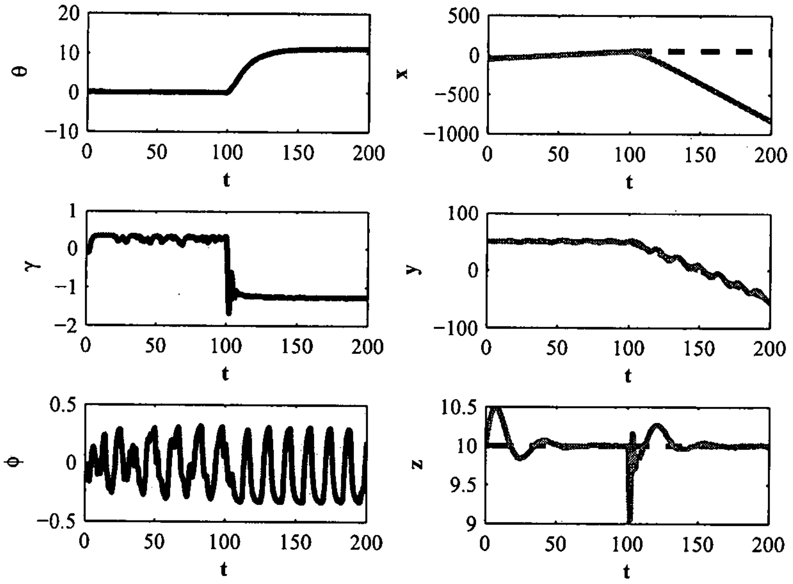


Figure 4.73: ALTAV states corresponding to a hard over fault in F_3 actuator

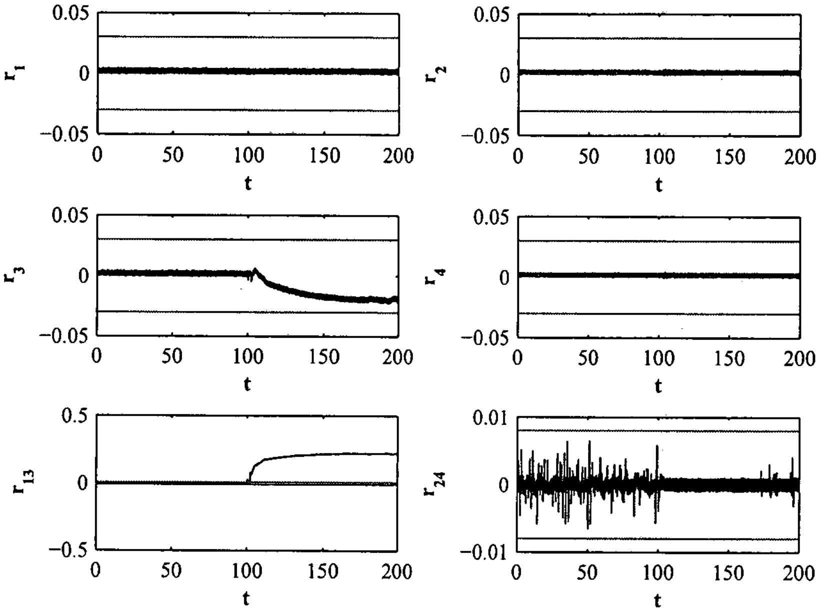


Figure 4.74: Residuals corresponding to a lock in place fault in F_3 actuator

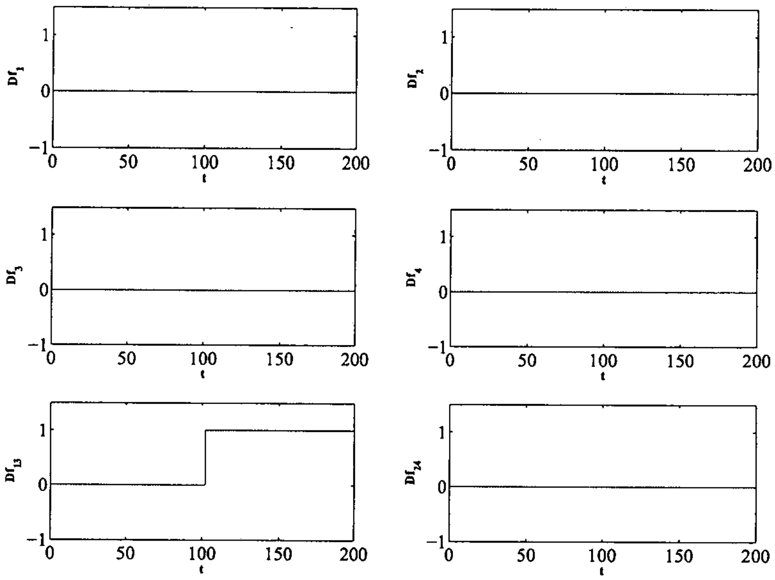


Figure 4.75: Detection flags corresponding to a lock in place fault in F_3 actuator

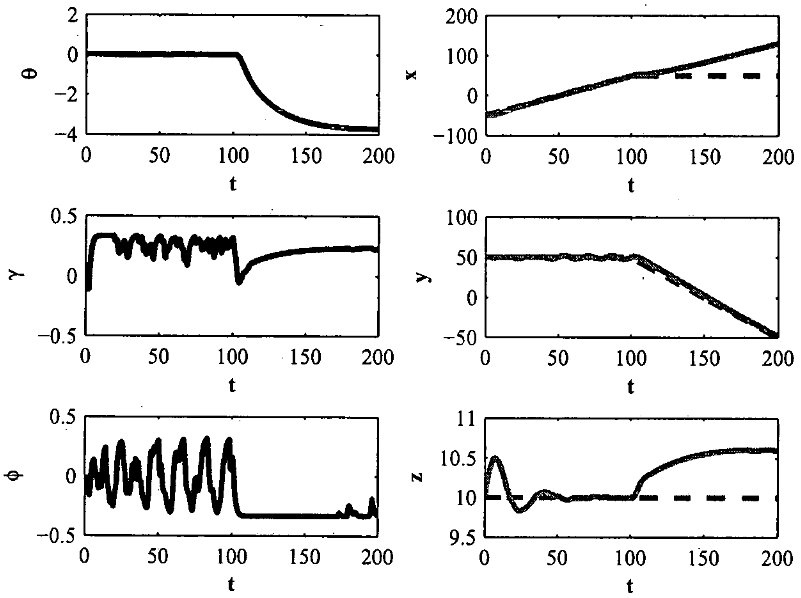


Figure 4.76: ALTAV states corresponding to a lock in place fault in F_3 actuator

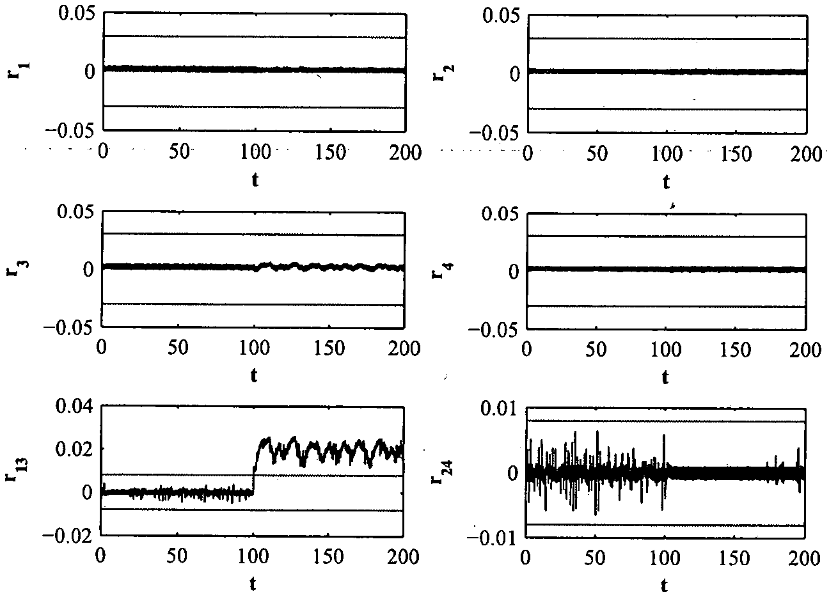


Figure 4.77: Residuals corresponding to a 20% loss of effectiveness fault in F_3 actuator

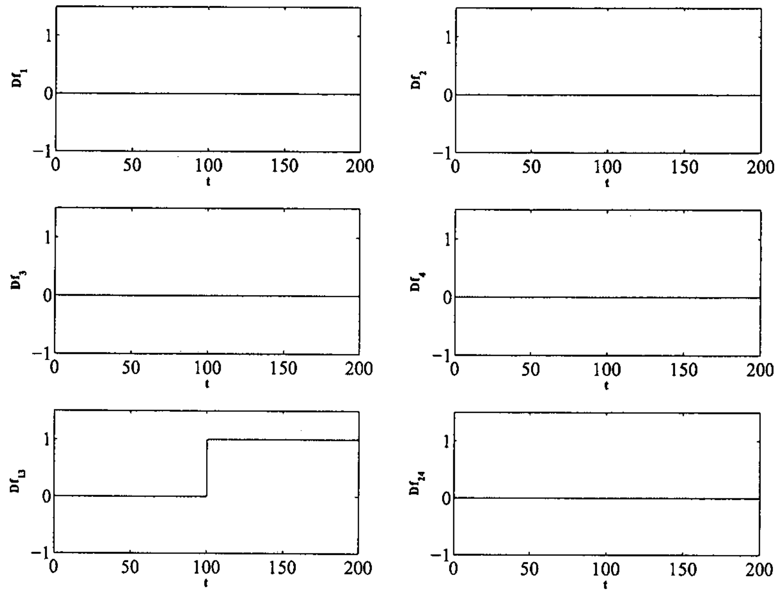


Figure 4.78: Detection flags corresponding to a 20% loss of effectiveness fault in F_3 actuator

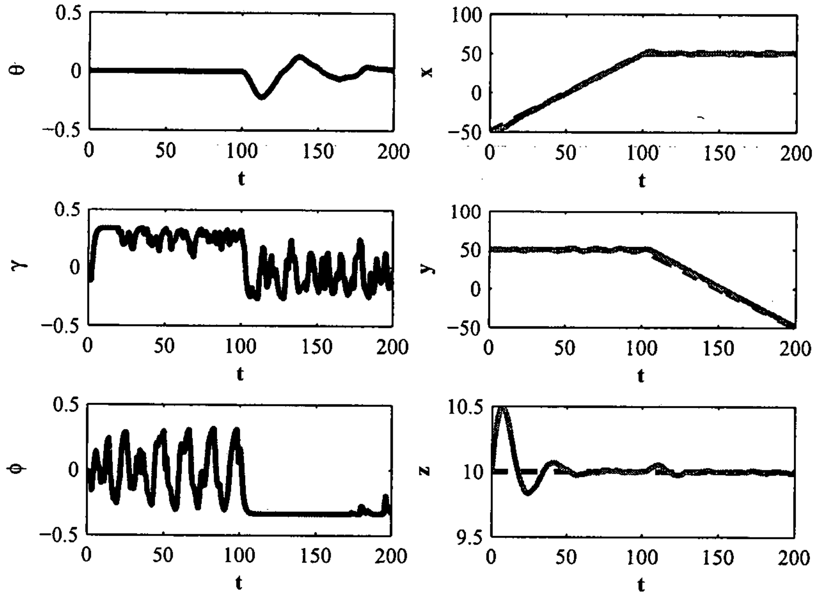


Figure 4.79: ALTAV states corresponding to a 20% loss of effectiveness fault in F_3 actuator

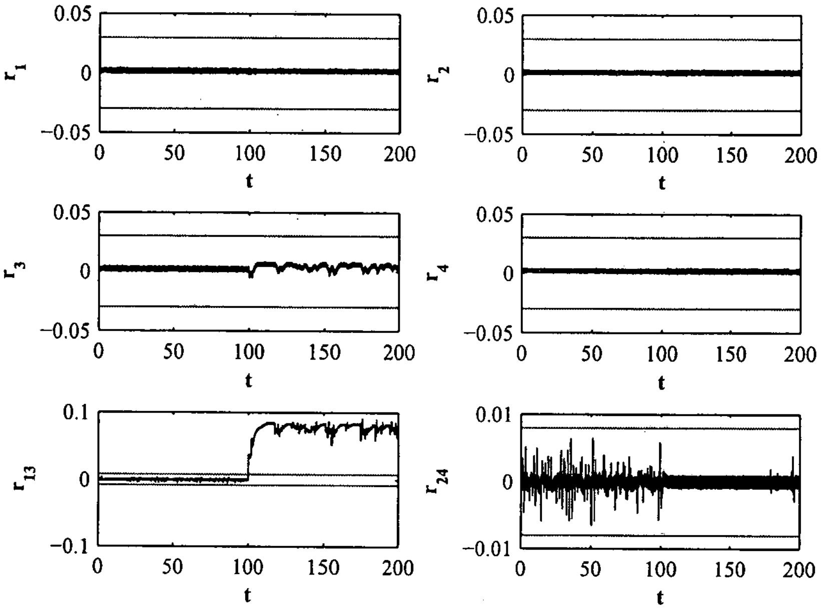


Figure 4.80: Residuals corresponding to a 50% loss of effectiveness fault in F_3 actuator

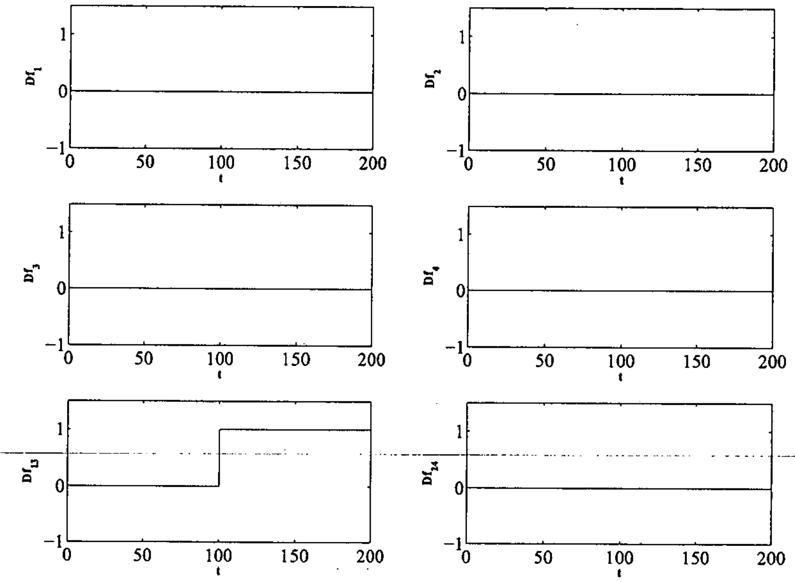


Figure 4.81: Detection flags corresponding to a 50% loss of effectiveness fault in F_3 actuator

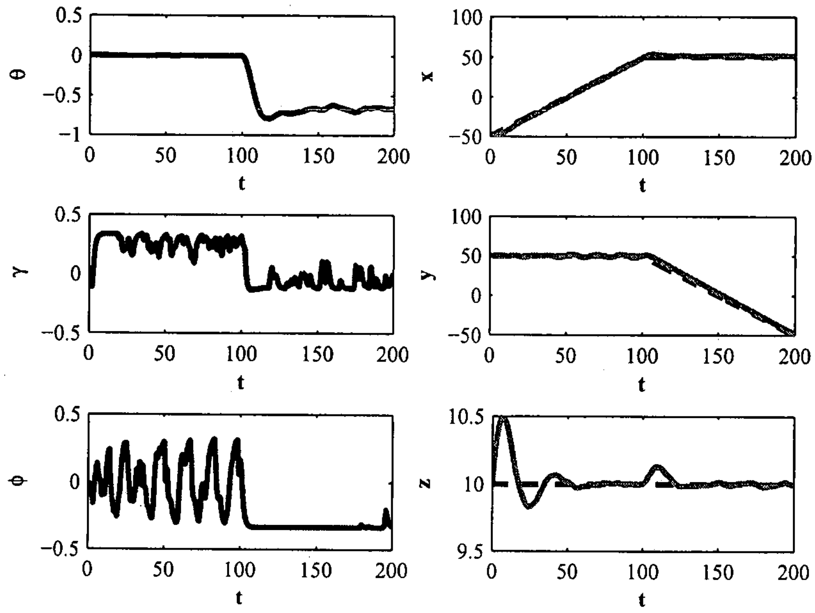


Figure 4.82: ALTAV states corresponding to a 50% loss of effectiveness fault in F_3 actuator

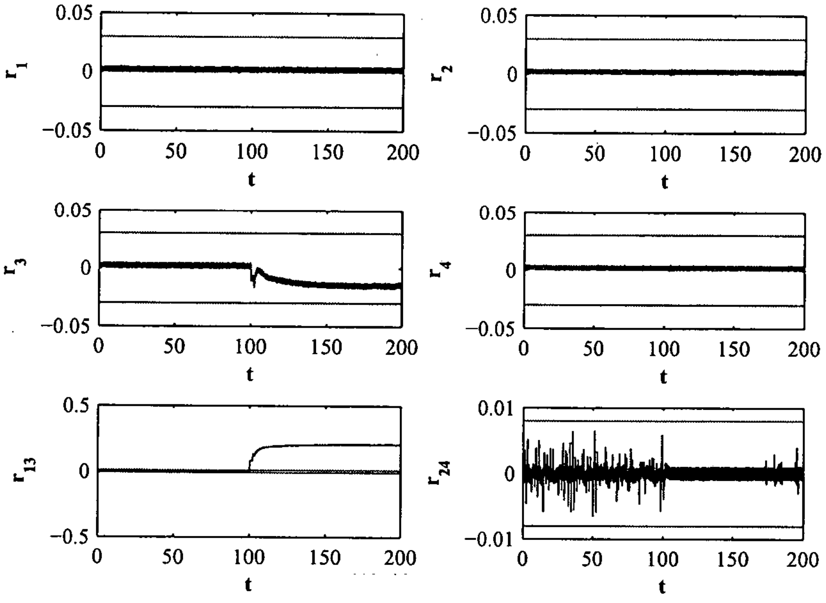


Figure 4.83: Residuals corresponding to a 80% loss of effectiveness fault in F_3 actuator

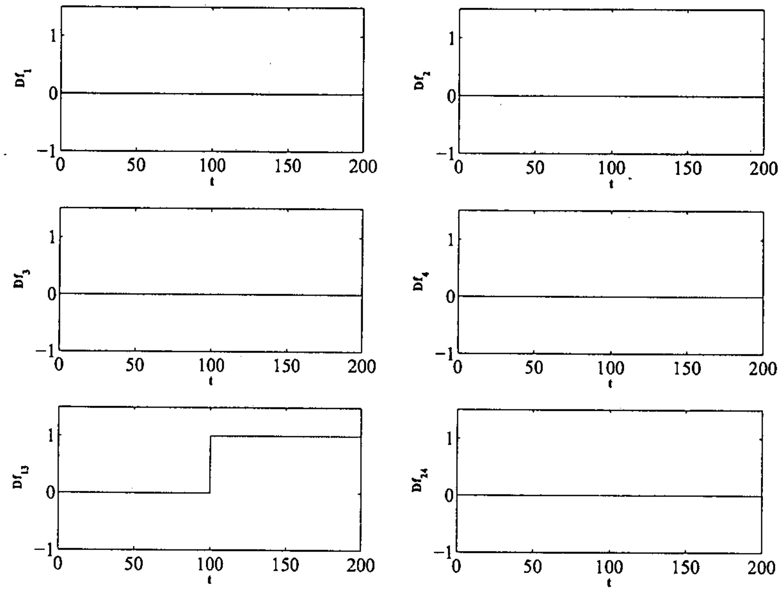


Figure 4.84: Detection flags corresponding to a 80% loss of effectiveness fault in F_3 actuator

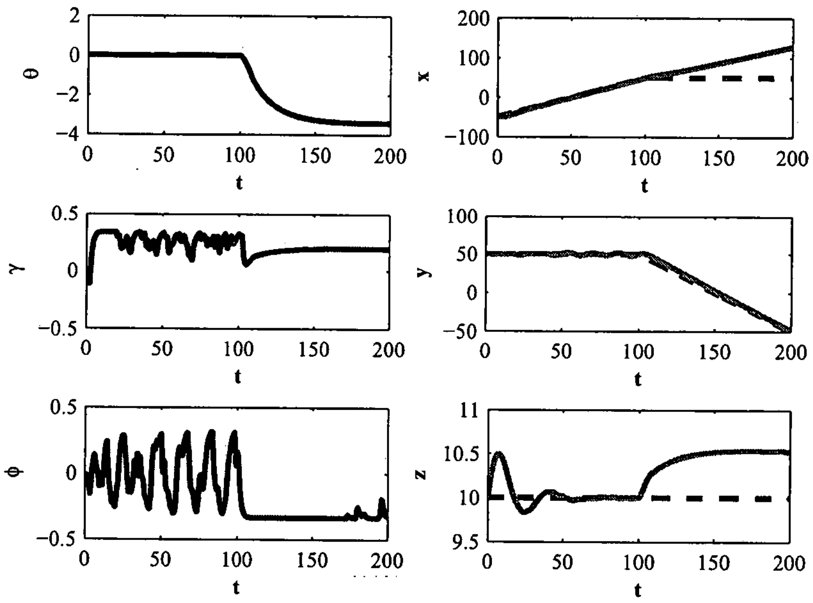


Figure 4.85: ALTAV states corresponding to a 80% loss of effectiveness fault in F_3 actuator

Type of fault	Detection	Isolation
Float	✓	✓
Hard over	✓	✓
Lock in place	✓	-
20% Loss of effectiveness	-	-
50% Loss of effectiveness	✓	-
80% Loss of effectiveness	✓	-

Table 4.12: Fault detection and isolation results for F_4 actuator

Figures 4.87, 4.88 and 4.86 show the residuals, the detection flags and ALTAV state correspond to a float fault in F_4 actuator at $t = 100$ seconds, respectively. As it is shown in figure 4.87, the residuals r_2 and r_{24} exceed their thresholds but after about 30 seconds the residual r_2 starts to oscillate around zero due to oscillations present in γ. However by considering the residual r_{24} which remains above its threshold we may conclude that the fault still exists in the input channel F_4. According to 4.88 one can detect and isolate the float fault in F_4. Figures 4.90, 4.91 and 4.89 depict the residuals and the detection flags correspond to a hard over fault in F_4 actuator at $t = 100$ seconds, respectively. In this faulty scenario too, although there are some oscillations around zero in the residual r_2, one can still detect and isolate this fault in F_4 actuator. Figures 4.93, 4.94 and 4.92 depict the residuals, the detection flags and ALTAV states correspond to a lock in place fault in F_4 actuator at $t = 100$ seconds, respectively. According to these figures, one can just detect but not isolate the fault in the input channels F_2 and F_4. Figures 4.96, 4.97 and 4.95 show the residuals, the detection flags and the ALTAV states correspond to a 20% loss of effectiveness in F_4 actuator at $t = 100$ seconds, respectively. According to these figures one can only detect the occurrence of the fault in one of the input channels F_2 and F_4 but cannot isolate between these two. Figures 4.99, 4.100 and 4.98 show the residuals, the detection flags and the ALTAV states correspond to a 50% loss of effectiveness in F_4 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_2 and F_4 but cannot isolate between these two. Figures 4.102 , 4.103 and 4.101 show the residuals , the detection flags and ATLAV states correspond to a 80% loss of effectiveness in F_4 actuator at $t = 100$ seconds, respectively. These figures demonstrate that one can only detect the occurrence of the fault in one of the input channels F_2 and F_4 but cannot isolate between these two.

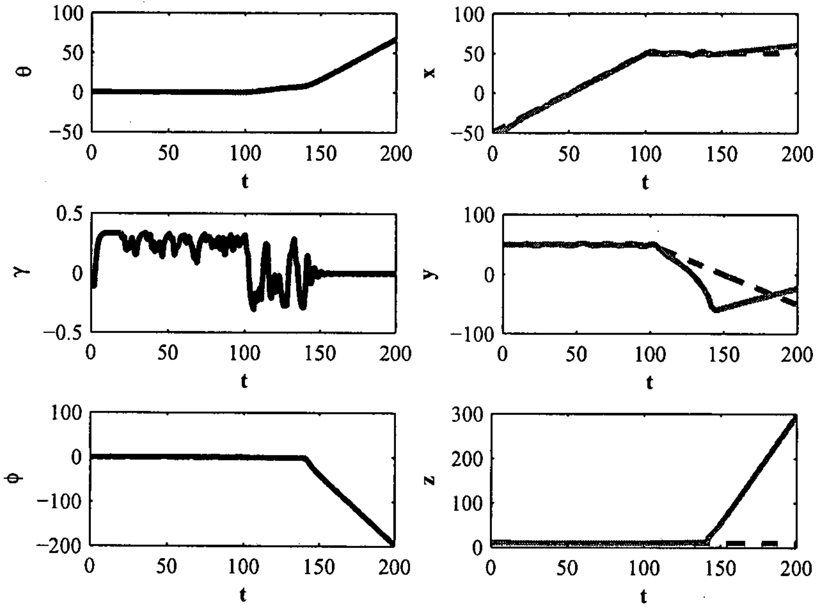


Figure 4.86: ALTAV states corresponding to a float fault in F_4 actuator

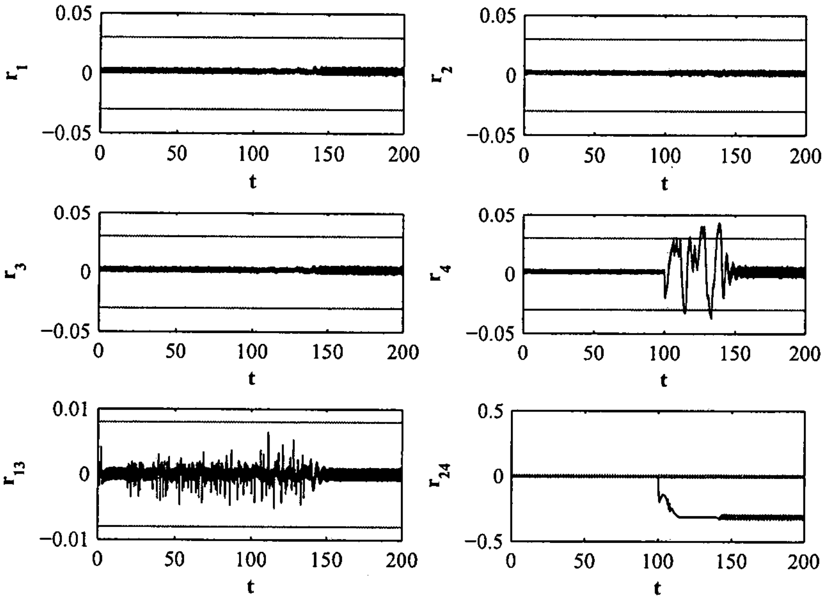


Figure 4.87: Residuals corresponding to a float fault in F_4 actuator

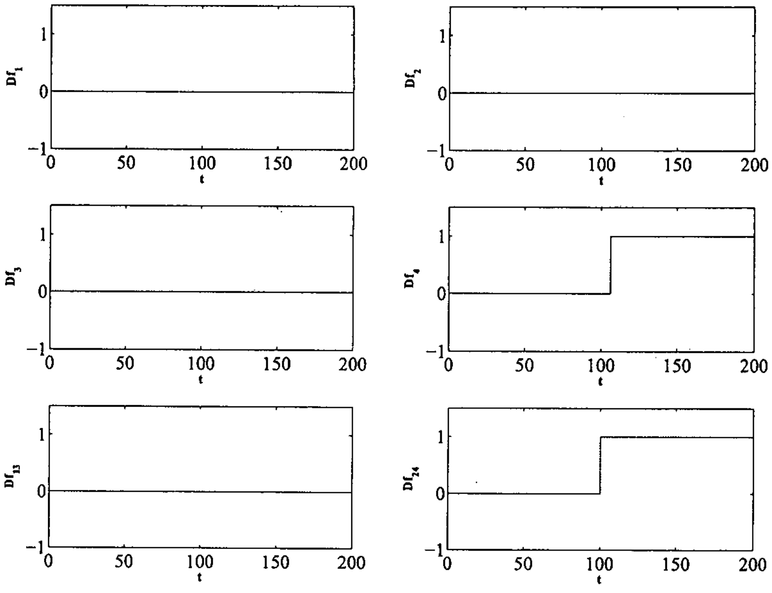


Figure 4.88: Detection flags corresponding to a float fault in F_4 actuator

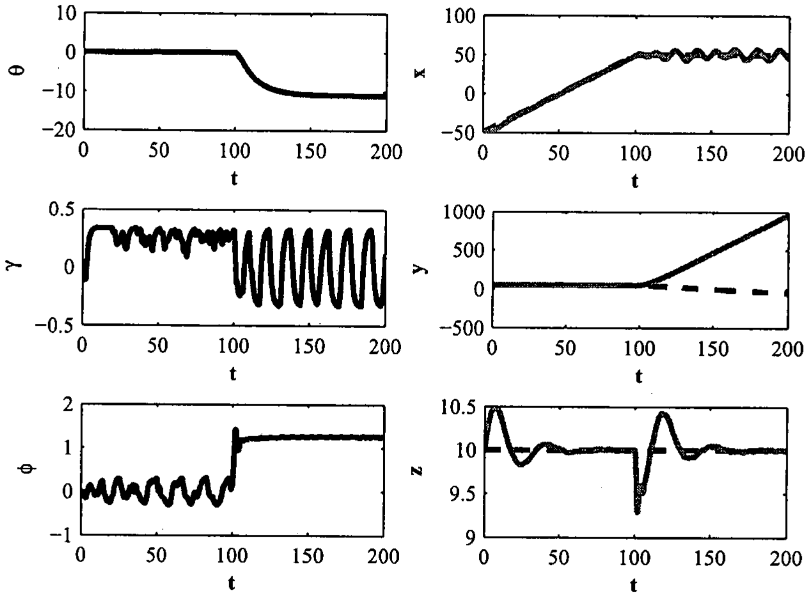


Figure 4.89: ALTAV states corresponding to a hard over fault in F_4 actuator

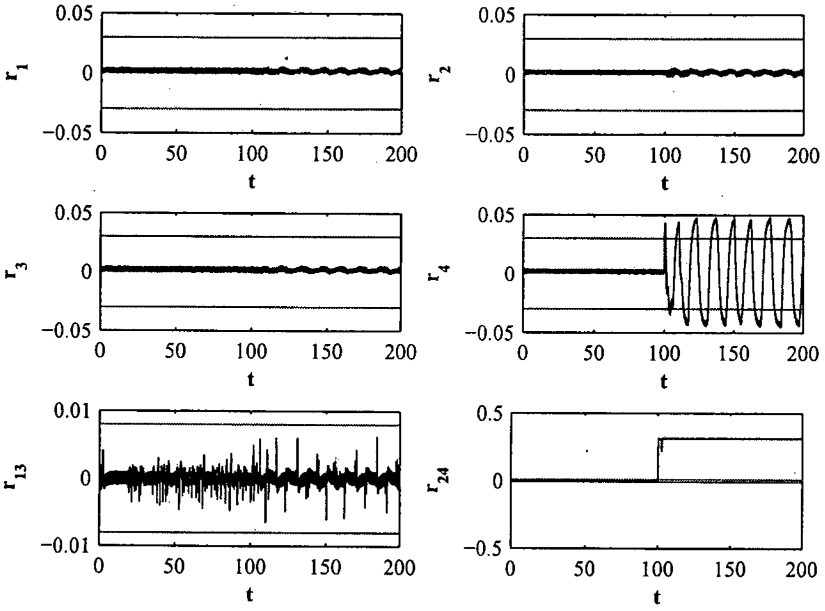


Figure 4.90: Residuals corresponding to a hard over fault in F_4 actuator

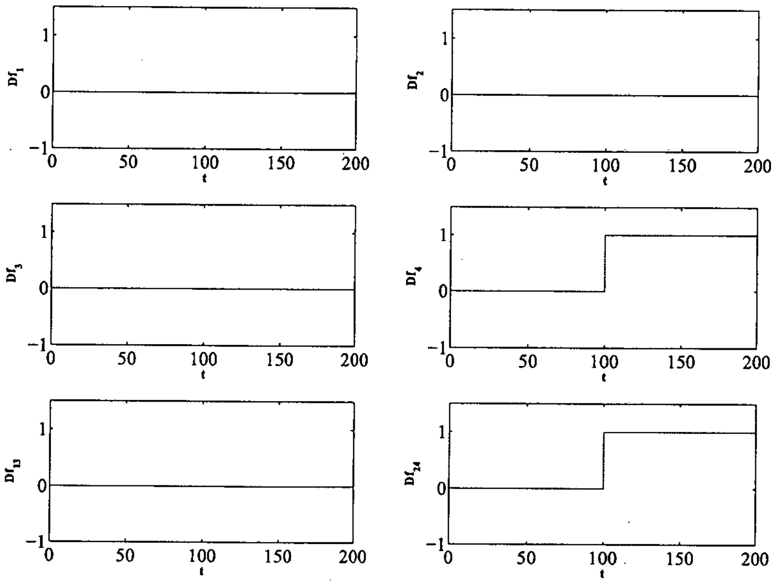


Figure 4.91: Detection flags corresponding to a hard over fault in F_4 actuator

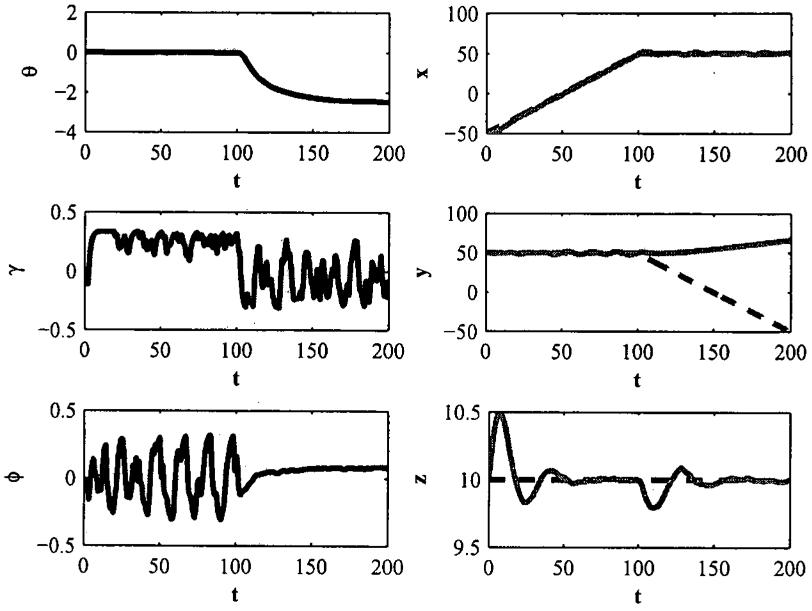


Figure 4.92: ALTAV states corresponding to a lock in place fault in F_4 actuator

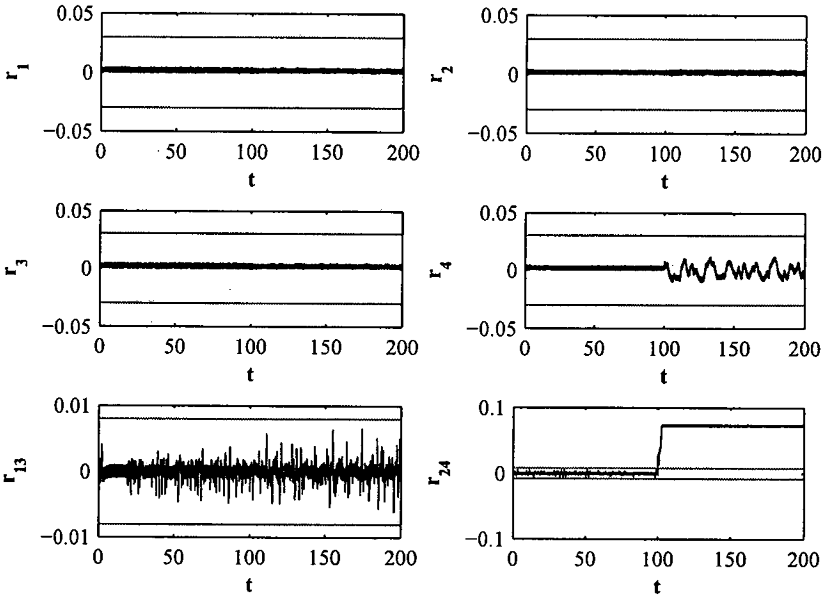


Figure 4.93: Residuals corresponding to a lock in place fault in F_4 actuator

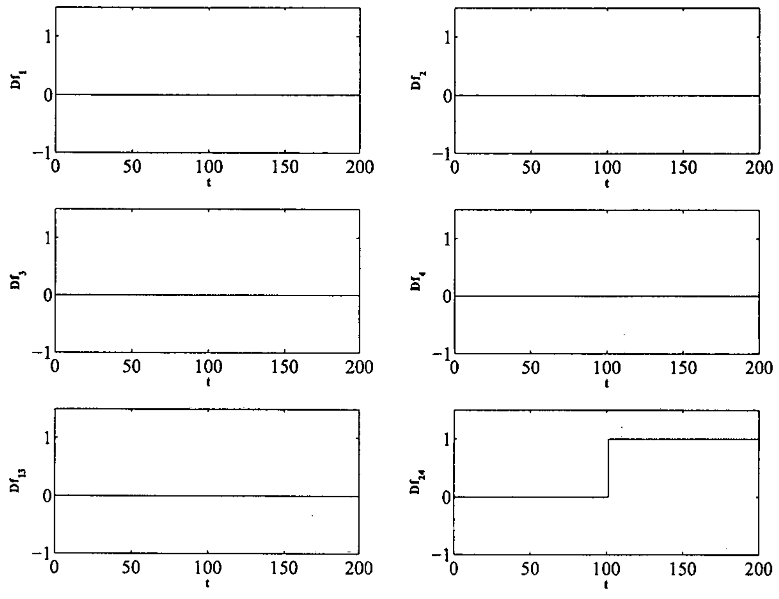


Figure 4.94: Detection flags corresponding to a lock in place fault in F_4 actuator

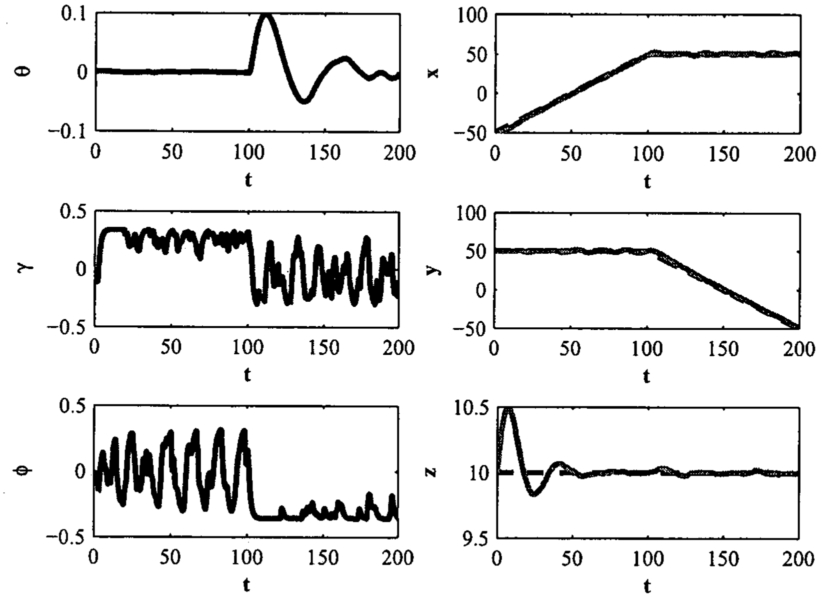


Figure 4.95: ALTAV states corresponding to a 20% loss of effectiveness fault in F_4 actuator

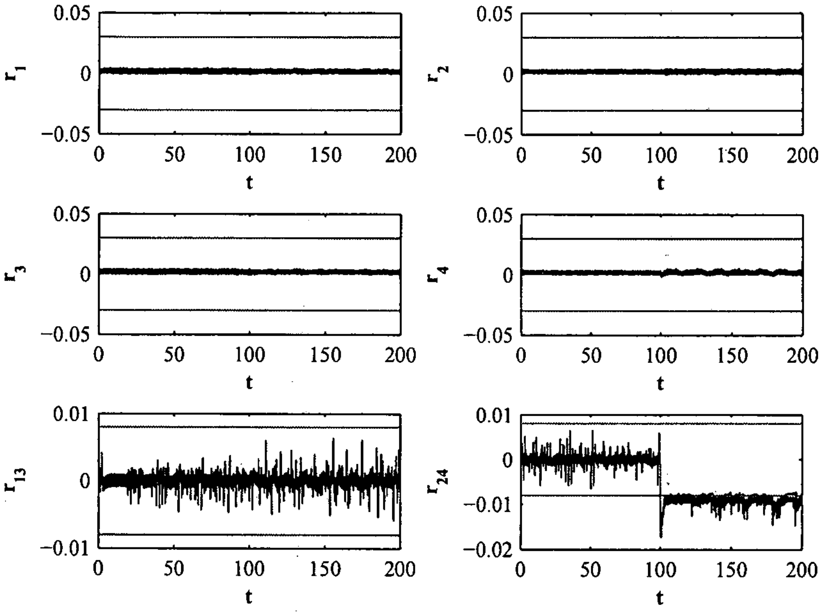


Figure 4.96: Residuals corresponding to a 20% loss of effectiveness fault in F_4 actuator

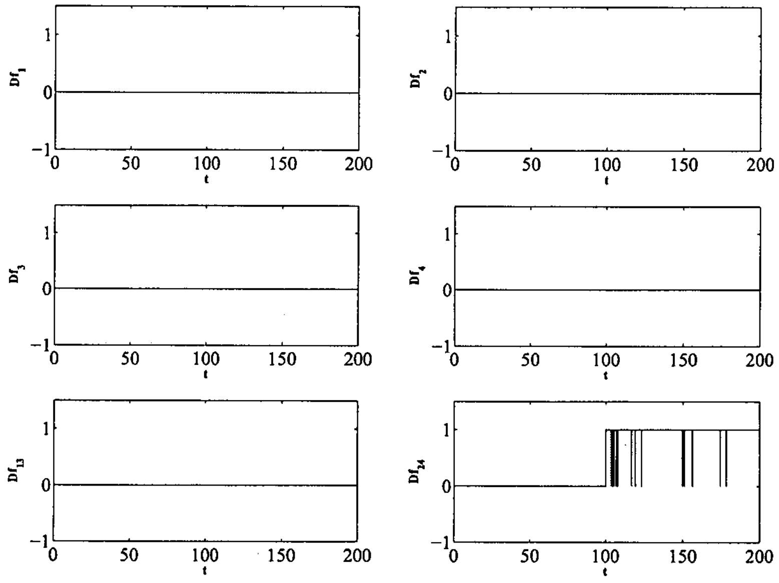


Figure 4.97: Detection flags corresponding to a 20% loss of effectiveness fault in F_4 actuator

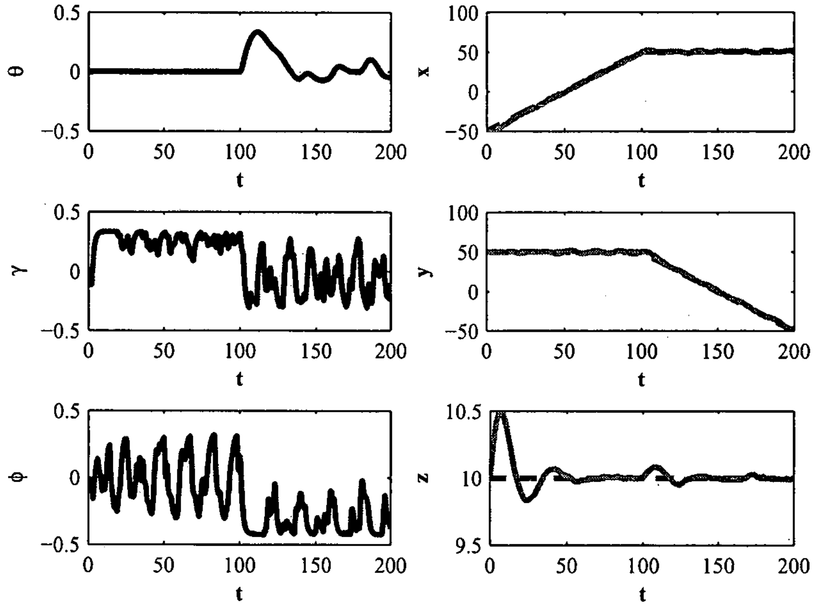


Figure 4.98: ALTAV states corresponding to a 50% loss of effectiveness fault in F_4 actuator

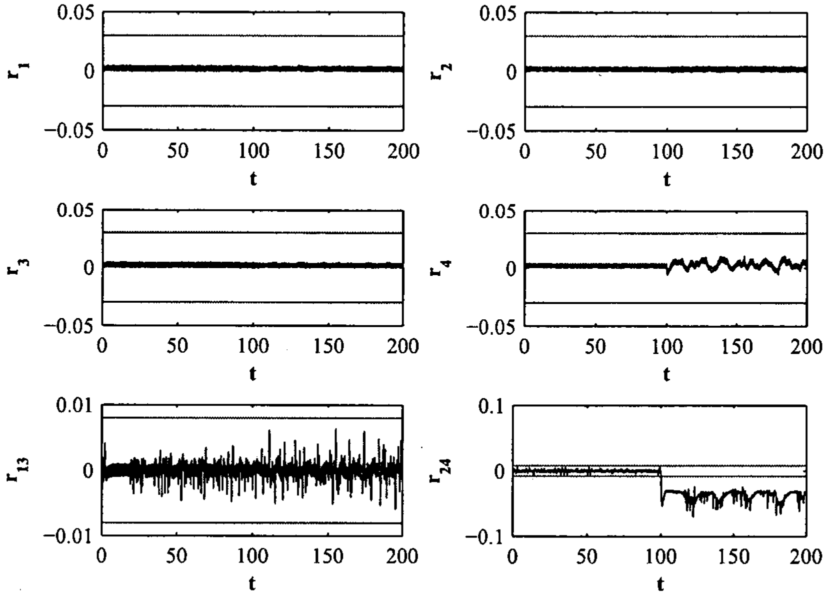


Figure 4.99: Residuals corresponding to a 50% loss of effectiveness fault in F_4 actuator

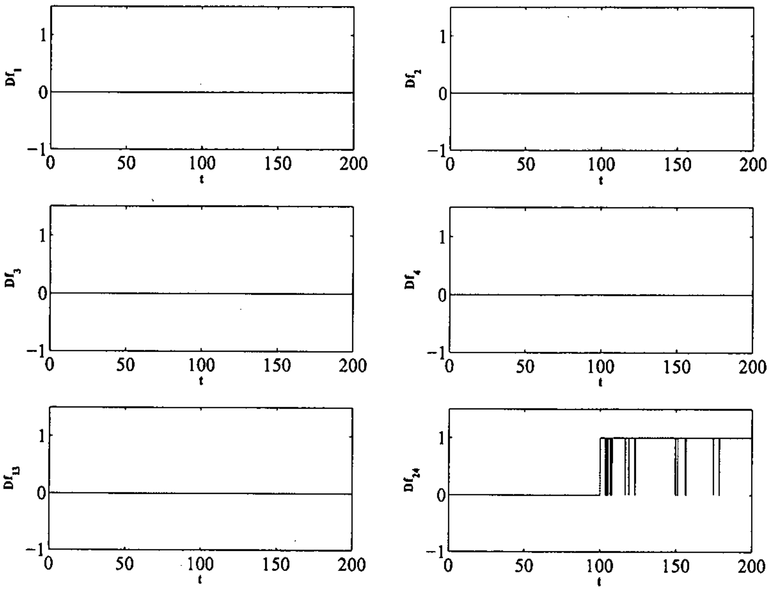


Figure 4.100: Detection flags corresponding to a 50% loss of effectiveness fault in F_4 actuator

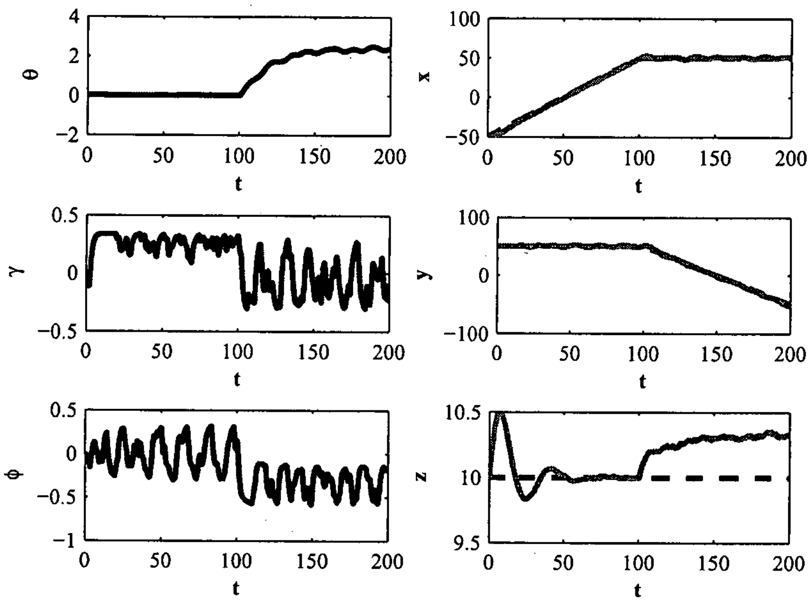


Figure 4.101: ALTAV states corresponding to a 80% loss of effectiveness fault in F_4 actuator

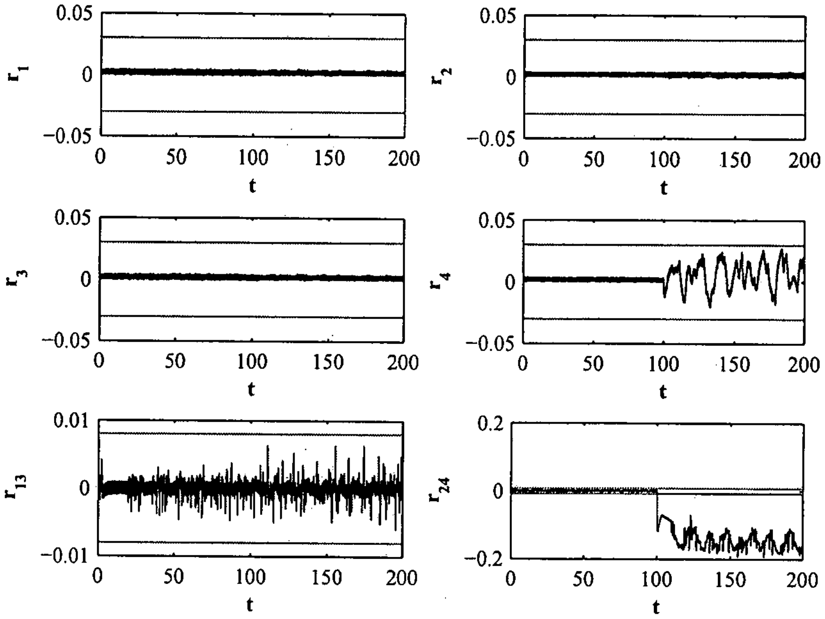


Figure 4.102: Residuals corresponding to a 80% loss of effectiveness fault in F_4 actuator

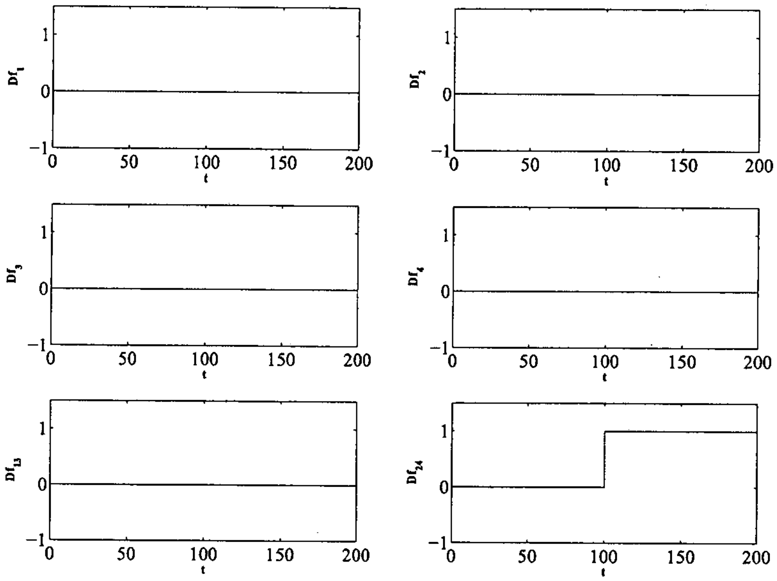


Figure 4.103: Detection flags corresponding to a 80% loss of effectiveness fault in F_4 actuator

4.3.6 Multiple Concurrent Faults Scenarios

In this section, the multiple concurrent fault scenarios will be considered and simulation results show the effectiveness of the proposed FDI method for detection and isolation of the concurrent faults. Figures 4.104, 4.105 and 4.106 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_1 actuator at $100 \leq t < 150$ seconds and the float fault in F_2 actuator at $t \geq 120$ seconds, respectively. As it is shown in these figures one can detect and isolate both faults in the input channels F_1 and F_2. Figures 4.107, 4.108 and 4.109 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_1 actuator at $100 \leq t < 150$ seconds and the float fault in F_3 actuator at $t \geq 120$ seconds, respectively. According to these figures, one can detect the occurrence of the float fault in F_1 actuators right after the occurrence of the fault ($t = 100$) but when the second fault happens in F_3 actuator, the residuals r_1, r_3 and r_{13} become zero after some transient and the detection flags does not shows the existence of these concurrent faults in the system until the time that the float fault in F_1 will be disappeared. Then the detection flags show the fault in F_3. This happens since in this scenario float fault has happened in both F_1 and F_3 actuators ($F_1 = 0, F_3 = 0$) and the controller command in this situation assigns the same value for both F_1 and F_3 input command which means that $F_1 - F_3 = 0$. Therefore the residual r_{13} is not affected in the scenario Since this residuals is only affected by the difference between F_1 and F_3.

Figures 4.110, 4.111 and 4.112 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_1 actuator at $100 \leq t < 150$ seconds and the hard over fault in F_3 actuator at $t = 120$ seconds, respectively. These figures shows that in this scenario one can detect and isolate both faults in F_1 and F_3 actuators. However, as it is mention before detection flags cannot detect the removal of the float fault from F_1 actuator . Figures 4.113, 4.114 and 4.115 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_1 actuator at $100 \leq t < 150$ seconds and a 50% loss of effectiveness in F_4 actuator at $t = 120$ seconds, respectively. In this scenario fault detection and isolation can be done perfectly by using the detection flags. Figures 4.116, 4.117 and 4.118 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_2 actuator at $100 \leq t < 150$ seconds and a float fault in F_4 actuator at $t = 120$ seconds, respectively. The same problem will happen also here (like the F_1,F_3 concurrent float faults) and the detection flags does not detect the fault in the time interval $120 < t < 150$ since the residual r_{24} is also affected only by the difference $F_2 - F_4$. Finally, Figures 4.116, 4.117 and 4.118 show the residuals, the detection flags and the ALTAV states correspond to a intermittent float fault in F_2 actuator at $100 \leq t < 150$ seconds and a hard over fault in F_4 actuator at $t = 120$ seconds, respectively. These figures shows that in this scenario one can detect and isolate both faults in F_2 and $F_4$3 actuators. However, as it is mention before detection flags cannot detect the removal of the float fault from F_2 actuator .

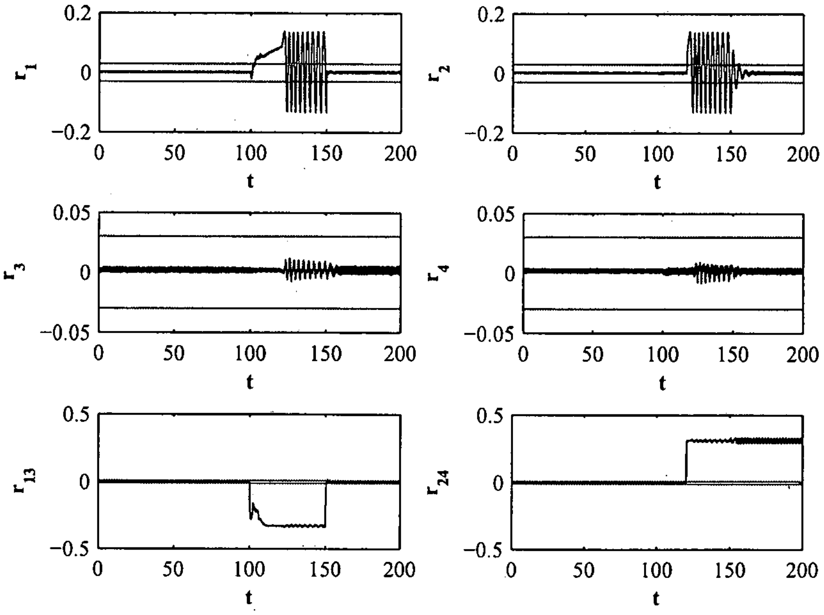


Figure 4.104: Residuals corresponding to concurrent float faults in F_1 and F_2 actuators

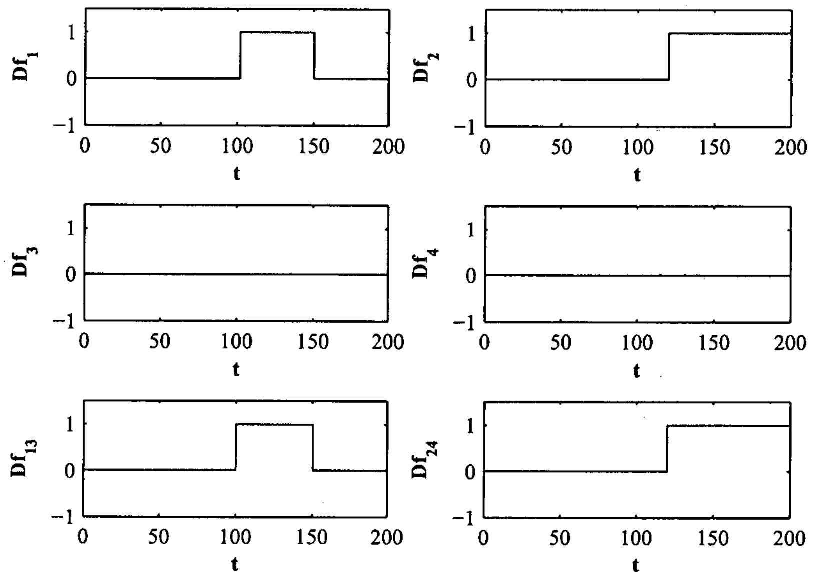


Figure 4.105: Detection flags corresponding to simultaneous float faults in F_1 and F_2 actuators

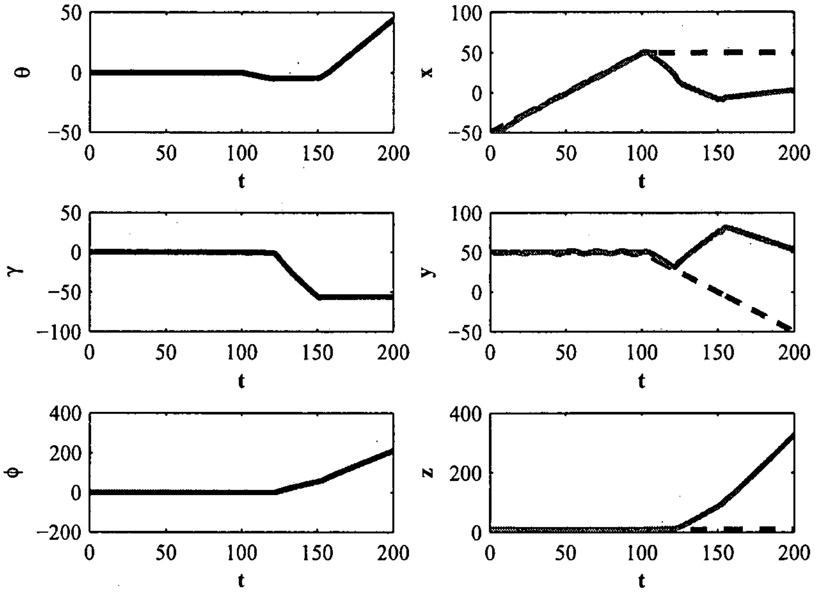


Figure 4.106: ALTAV states corresponding to concurrent float faults in F_1 and F_2 actuators

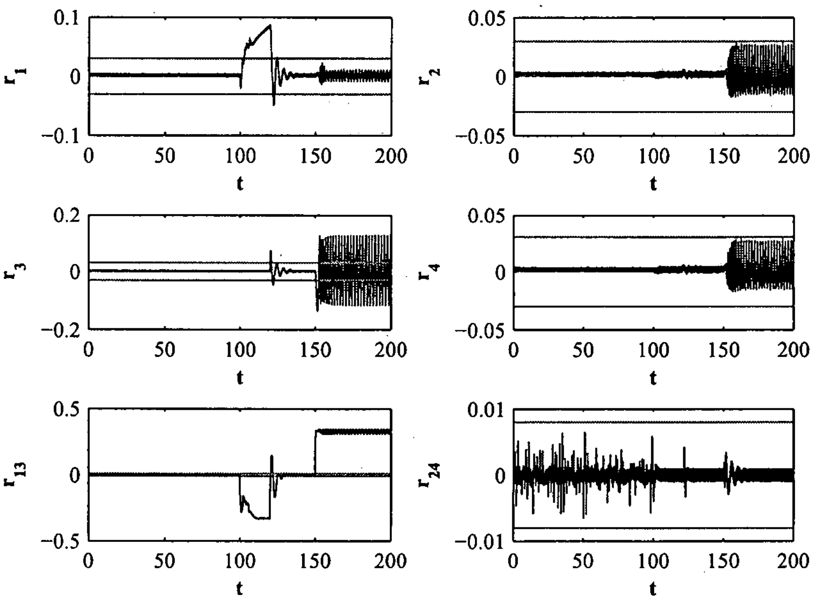


Figure 4.107: Residuals corresponding to concurrent float faults in F_1 and F_3 actuators

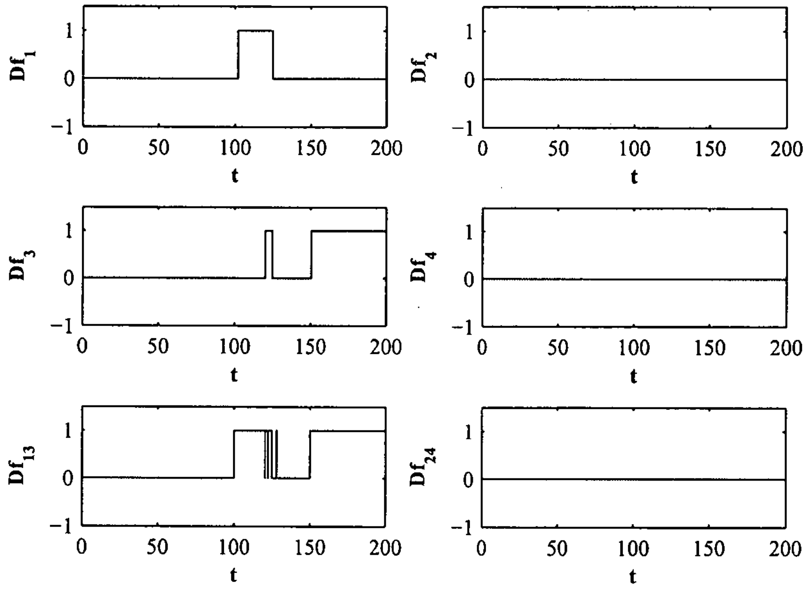


Figure 4.108: Detection flags corresponding to concurrent float faults in F_1 and F_3 actuators

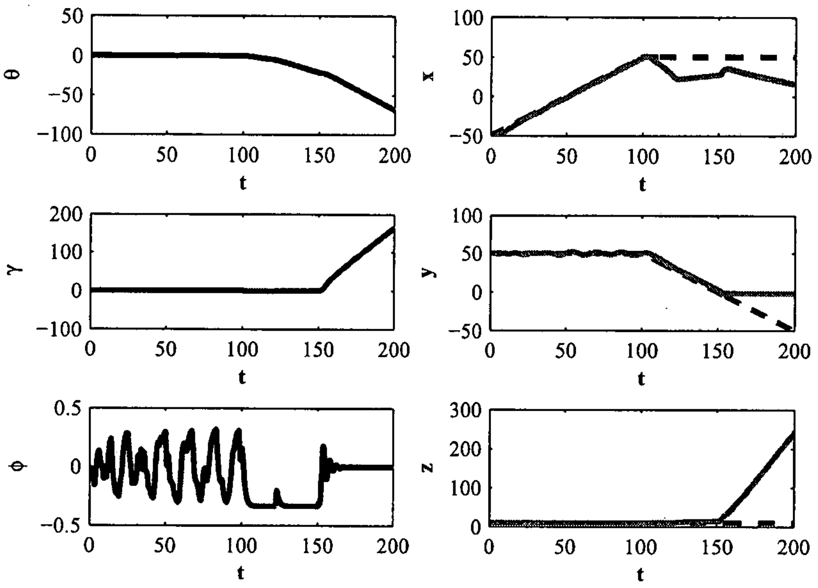


Figure 4.109: ALTAV states corresponding to concurrent float faults in F_1 and F_3 actuators

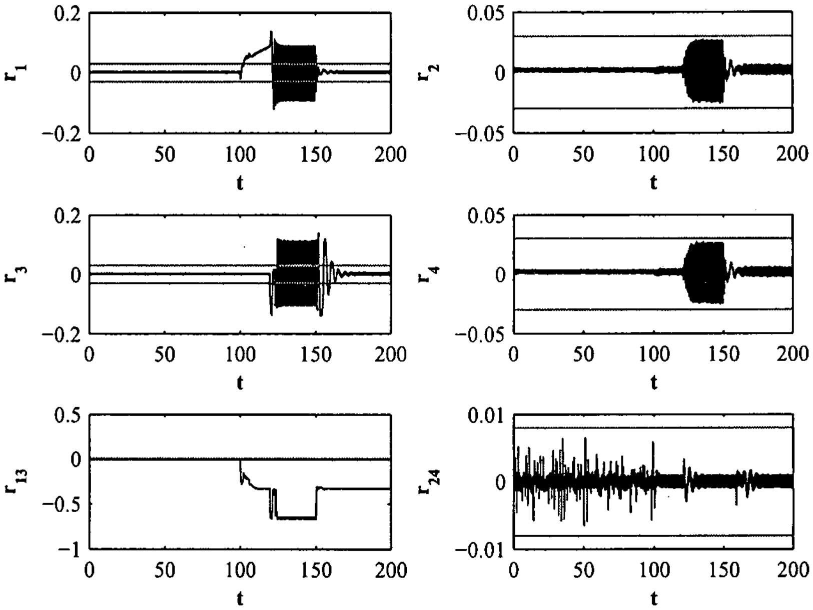


Figure 4.110: Residuals corresponding to concurrent faults in F_1 and F_3 actuators (F_1: Float , F_3: Hard over)

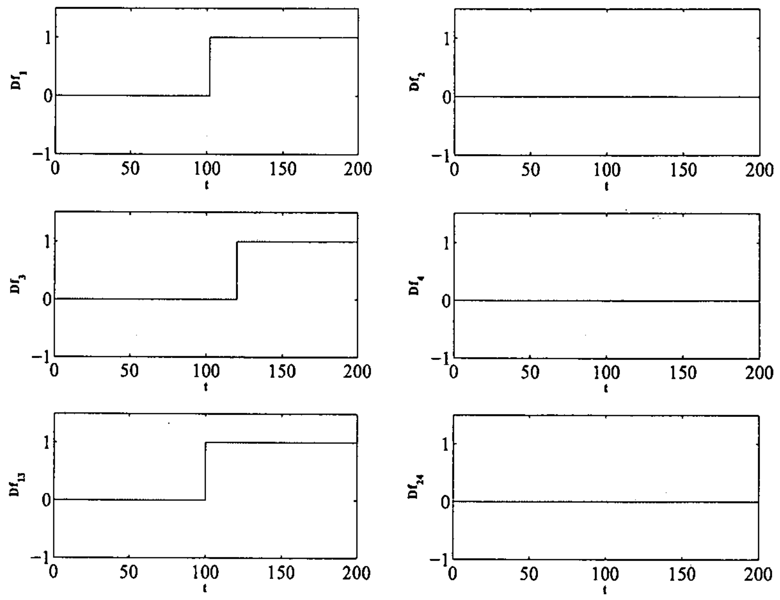


Figure 4.111: Detection flags corresponding to concurrent faults in F_1 and F_3 actuators (F_1: Float, F_3: Hard over)

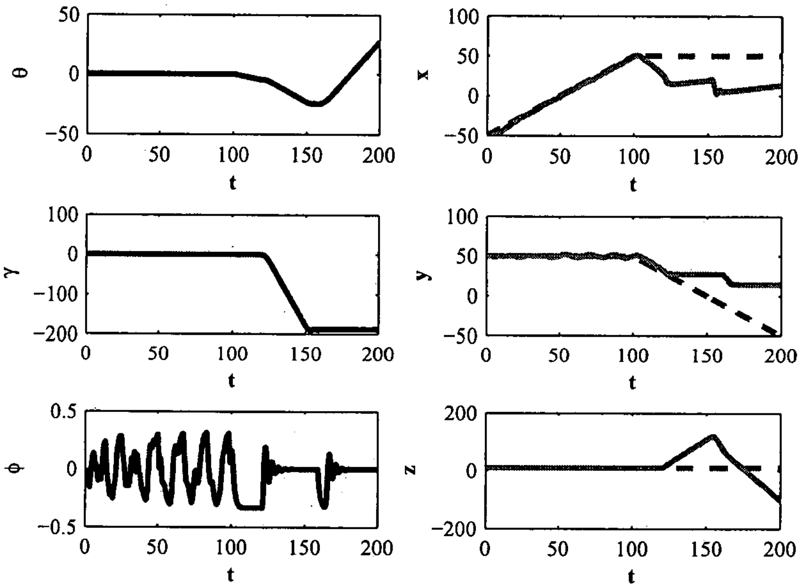


Figure 4.112: ALTAV states corresponding to concurrent faults in F_1 and F_3 actuators (F_1: Float , F_3: Hard over)

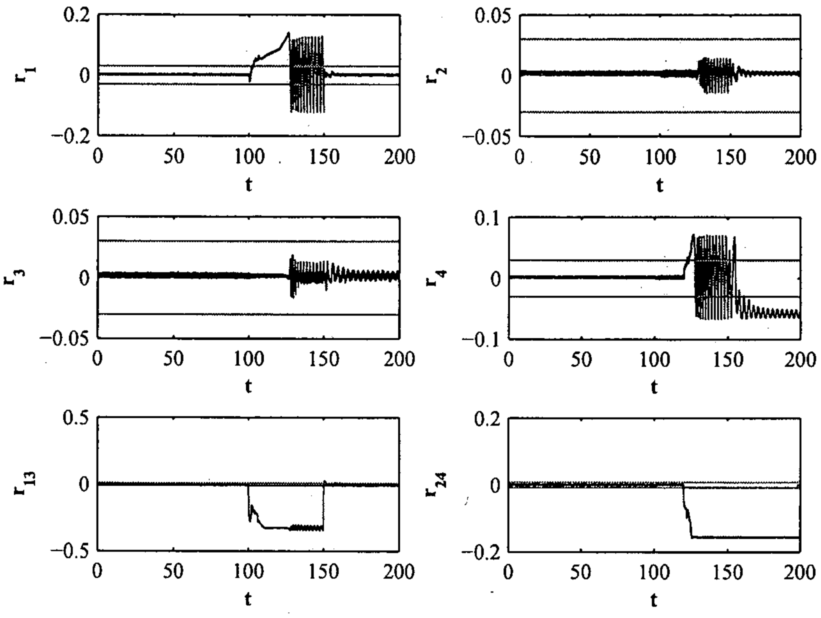


Figure 4.113: Residuals corresponding to concurrent faults in F_1 and F_4 actuators (F_1: Float , F_4: 50% Loss of effectiveness)

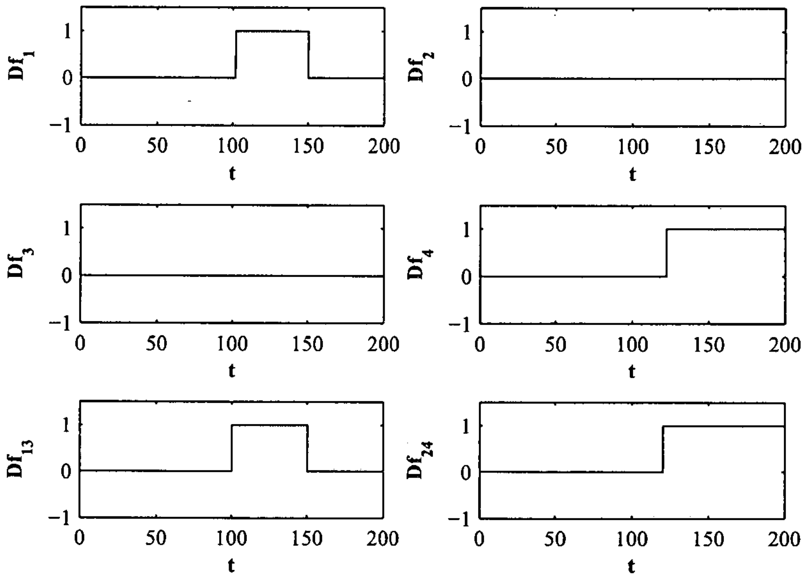


Figure 4.114: Detection flags corresponding to concurrent faults in F_1 and F_4 actuators (F_1: Float , F_4: 50% Loss of effectiveness)

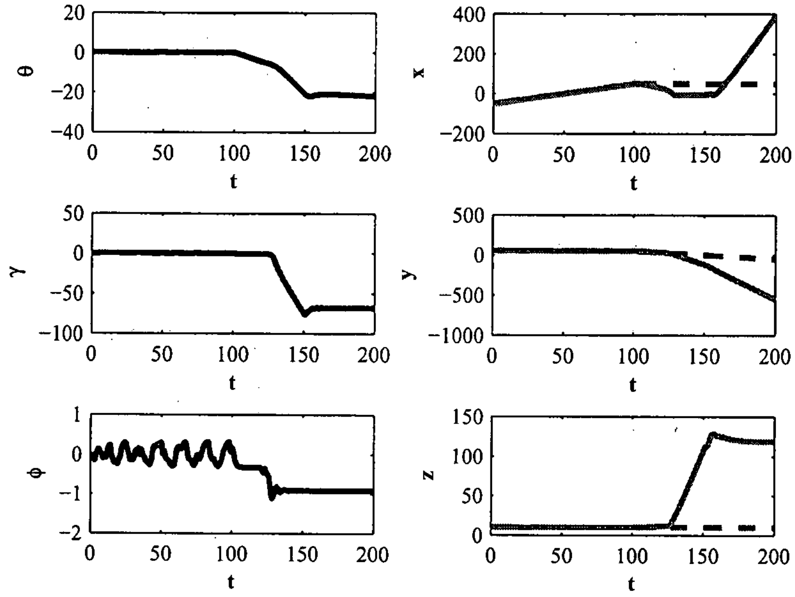


Figure 4.115: ALTAV states corresponding to concurrent faults in F_1 and F_4 actuators (F_1: Float , F_4: 50% Loss of effectiveness)

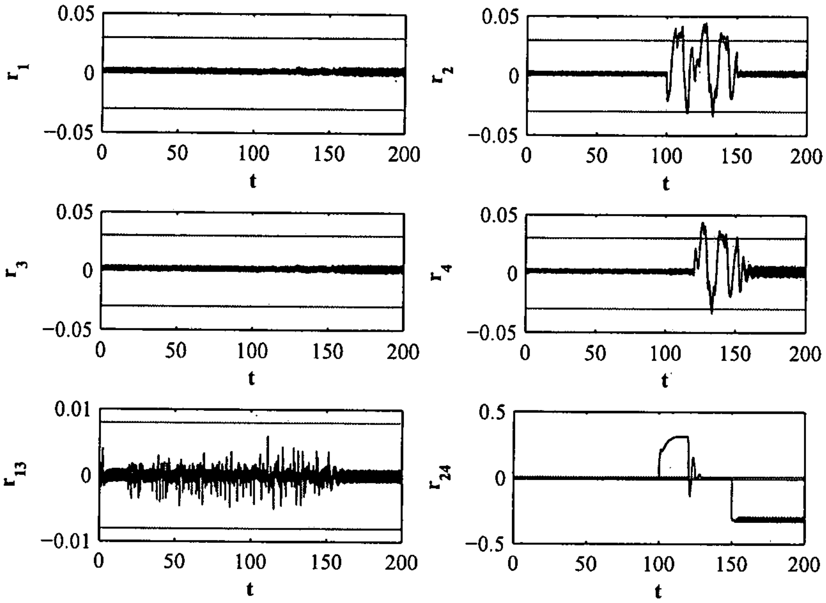


Figure 4.116: Residuals corresponding to concurrent float faults in F_2 and F_4 actuators

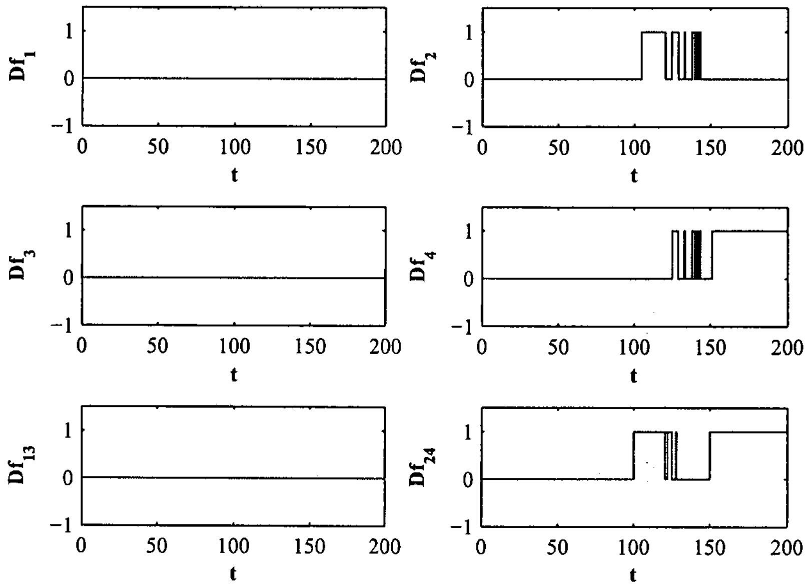


Figure 4.117: Detection flags corresponding to concurrent float faults in F_2 and F_4 actuators

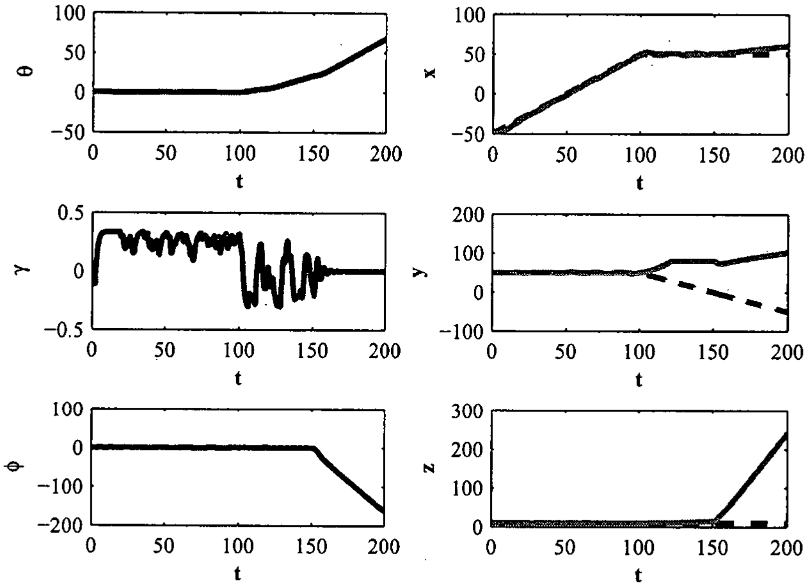


Figure 4.118: ALTAV states corresponding to concurrent float faults in F_2 and F_4 actuators

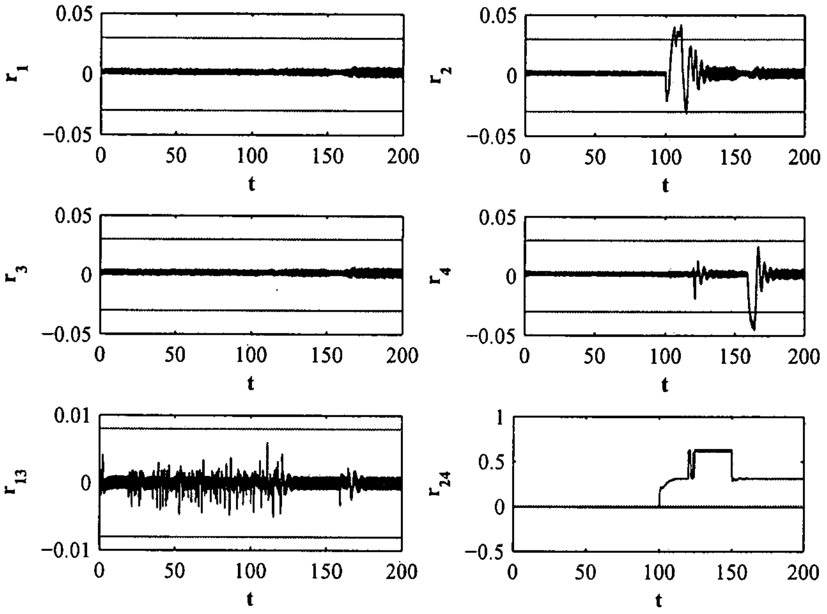


Figure 4.119: Residuals corresponding to concurrent faults in F_2 and F_4 actuators (F_2: Float , F_4: Hard over)

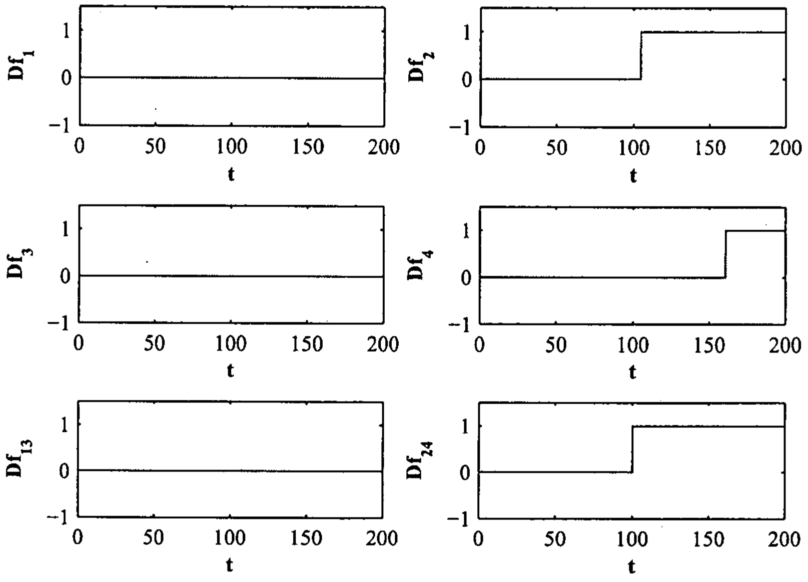


Figure 4.120: Detection flags corresponding to concurrent faults in F_2 and F_4 actuators (F_2: Float , F_4: Hard over)

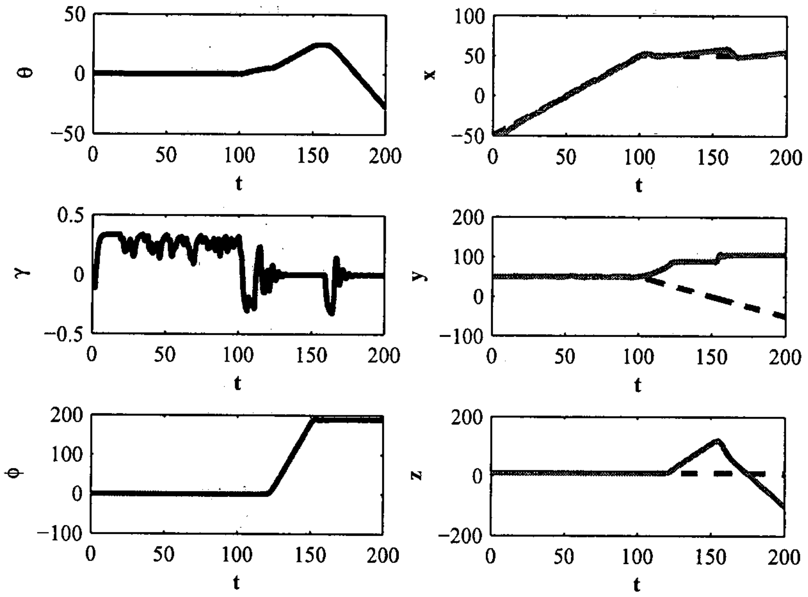


Figure 4.121: ALTAV states corresponding to concurrent faults in F_2 and F_4 actuators (F_2: Float , F_4: Hard over)

4.3.7 Safe mode operation for ALTAV

In this section, fault recovery system will be investigated for the ALTAV system. A fault tolerant control systems needs to recover from the faults after the fault detection and isolation (FDI). In the previous section, the required detection filters for FDI was proposed. However, for fault recovery one should identify the type of fault too. In order to warranty the safety of the vehicle motion during the fault diagnosis procedure, one should switch the controller to a safe mode operation. According to the dynamics of ALTAV, the input channels may be divided into two pairs, 1) F_1, F_3, 2) F_2, F_4. First we consider that the fault has been detected in one of the input channels of the first pair. In this case, we should turn off both actuators F_1 and F_3 and apply the same command for the input channels F_2 and F_4 ($F = F_2 = F_4$). The equation of ALTAV motion in this scenario can be written as follows:

$$\begin{aligned}
M\ddot{x} &= 2Fsin(\gamma) - C_x\dot{x} \\
M\ddot{y} &= 2Fsin(\phi) - C_y\dot{y} \\
M\ddot{z} &= -2Fcos(\gamma)cos(\phi) - F_B + Mg - C_z\dot{z} \\
J_\theta\ddot{\theta} &= -2Fsin(\rho) - C_\theta\dot{\theta} \\
J_\gamma\ddot{\gamma} &= -F_BL_Bsin(\gamma) - C_\gamma\dot{\gamma} \\
J_\phi\ddot{\phi} &= -F_BL_Bsin(\phi) - C_\phi\dot{\phi}
\end{aligned} \tag{4.48}$$

We can rewrite the dynamics of γ in this condition as

$$\ddot{\gamma} + \frac{C_\gamma}{J_\gamma}\dot{\gamma} + \frac{F_BL_B}{J_\gamma}sin(\gamma) = 0 \tag{4.49}$$

and by using Lyapunov stability theory we can prove that the origin $\gamma = 0$ is asymptotically stable for all initial value $\gamma_0 < \pi$. The same also holds for the ϕ dynamics. Therefore both γ and ϕ converge to zero. The equation of the translational dynamics of the ALTAV can be written as follows for $\gamma = 0$ and $\phi = 0$:

$$M\ddot{x} = -C_x\dot{x} \tag{4.50}$$

$$M\ddot{y} = -C_y\dot{y} \tag{4.51}$$

$$M\ddot{z} = -2F - F_B + Mg - C_z\dot{z} \tag{4.52}$$

According to the equation 4.50, the motion in x direction is stable and x converges to a fixed value and $\dot{x}$ converges to zero. The same also holds for the y direction. In the other words, the (x,y) position of the ALTAV converges to the fixed point in the (x,y) plane. We can use the existing freedom in the remaining input channels ($F_2 = F_4 = F$) to control the altitude of the vehicle by designing the controller for the z dynamics 4.52. The heading dynamics θ for the fixed controller input F (fixed set point for altitude)is governed by following equation

$$J_\theta\ddot{\theta} = -2Fsin(\rho) - C_\theta\dot{\theta} \tag{4.53}$$

which has the solution $\theta = at$ and $\dot{\theta} = a$ for some fixed value $a > 0$. This means that the vehicle rotates around the z axis in the safe mode operation. Figure 4.122 show the ALTAV

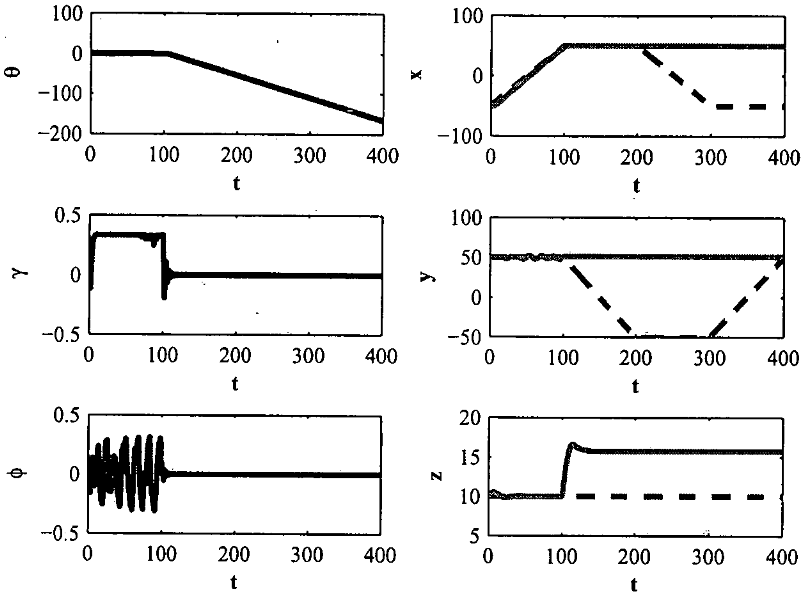


Figure 4.122: ATLAV states corresponding to a float fault in F_1 actuator and switching to safe mode

states corresponding to the float fault in F_1 actuator and control switching to the safe mode. As it is shown in this figure, γ and ϕ converge to zero, x and y coordinates converge to the fixed point and one can control the altitude. The same safe mode operation can be considered for the case when there is a fault in the input channels F_2 and F_4 by using the input channels F_1 and F_3.

Fault detection in the safe mode operation

In this part, we investigate the fault detection in the safe mode operation. As mentioned before, in the safe mode operation, the input channels are divided in two groups F_1, F_3 and F_2, F_4 and the the input channels in the faulty pair are set to zero. Moreover, the other input channels are used to control the altitude of the vehicle. The same residuals r_13 and r_24 can be used in the safe mode operation to detect the fault in these two group. we consider the scenario where the float fault happens in F_1 actuator at t = 100 sec. and the controller switch to the safe mode operation after detection of this fault. Then at t = 120 second the float fault will happen in F_4 actuator. Figure 4.123 shows the residual corresponding to scenario. As shown in this figure, one can easily detect the occurrence of the second fault in F_4 in the safe mode operation. It should be noted that after switching to the safe mode operation, the residual corresponding to the faulty pairs will go back to zero since the input channels in that pair are set to zero. Figure 4.124 depicts the ALTAV states corresponding to this fault scenario.

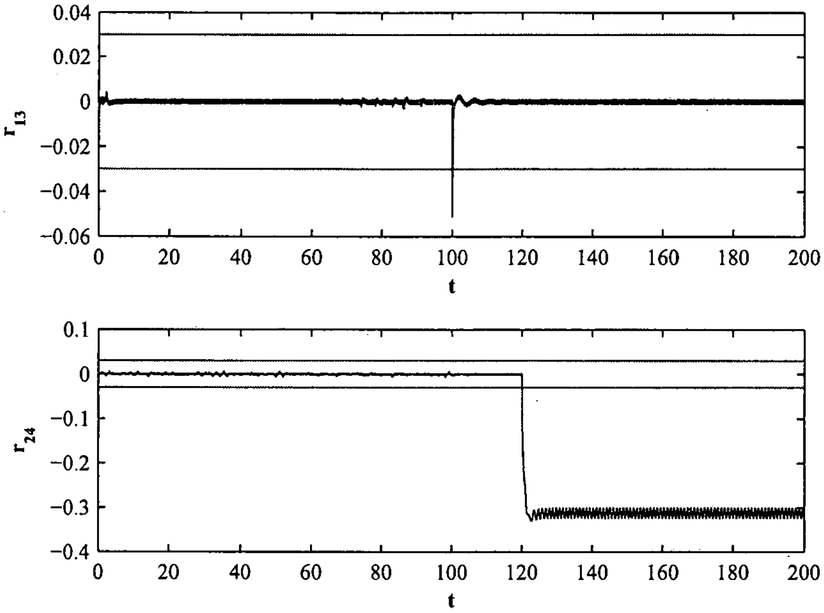


Figure 4.123: Safe mode operation residuals corresponding to a float fault in F_1 and F_4 actuators

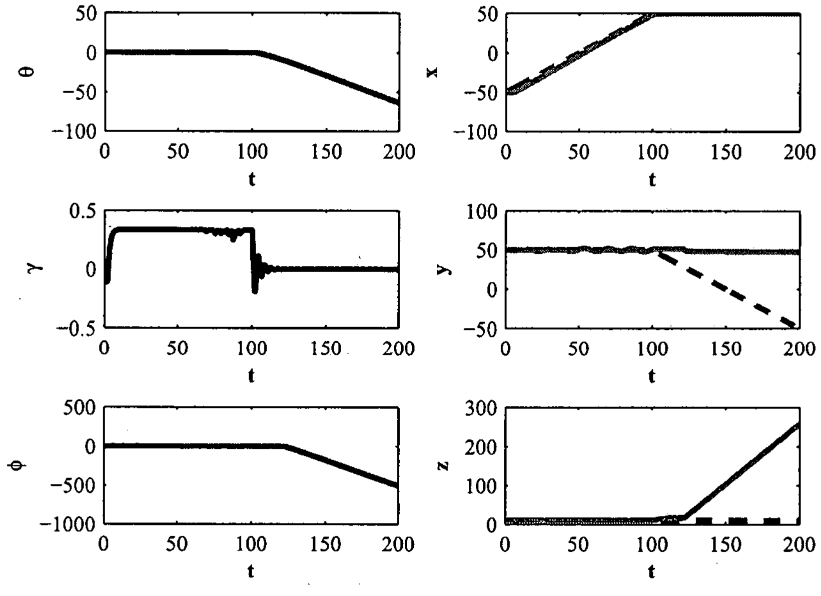


Figure 4.124: ATLAV states corresponding to a float fault in F_1 and F_4 actuators

4.3.8 Fault Diagnosis for ALTAV

In this section, fault diagnosis algorithms will be investigated for ALTAV system. As it is mentioned in the previous section, after detection of the fault, the controller will be switched to the safe mode controller in order to maintain the safety of the vehicle. In the safe mode operation, the fault diagnosis subsystem will be activated in order to identify the type of fault by injecting some auxiliary signals to the system.

Common actuator faults that are considered in this work include [?]: (i) freezing or lock in-place (LIP) fault, (ii) float fault, (iii) hard-over fault (HOF), and (iv) loss of effectiveness (LOE) fault. In the case of LIP fault, the actuator freezes at a particular situation and does not respond to subsequent commands. HOF is characterized by the actuator moving to its upper or lower position limits regardless of the commanded signal. The actuator speed of response is bounded by the actuator rate limits. Float fault occurs when the actuator floats with zero moment and does not contribute to the control authority. Loss of effectiveness is characterized by lowering the actuator gain with respect to its nominal value. Different types of actuator faults may be mathematically parameterized as follows:

$$u_{true} = \begin{cases} u_{cmd} & \text{No Failure Case} \\ k(t)u_{cmd} & 0 < \varepsilon \leq k(t) < 1, \forall t \geq t_F(\text{LOE}) \\ 0 & \forall t \geq t_F(\text{Float}) \\ u_{cmd}(t_F) & \forall t \geq t_F(\text{LIP}) \\ u_m \quad \text{or} \quad u_M & \forall t \geq t_F(\text{HOF}) \end{cases} \tag{4.54}$$

where t_F denotes the time instant of fault occurrence in the actuator, k denotes its effectiveness coefficient such that $k \in [\varepsilon, 1]$ and ε denotes its minimum effectiveness, and u_m and u_M denote the minimum and maximum value of the input respectively. The following general model

$$u_{true} = \sigma k u_{cmd} + (1 - \sigma)\bar{u} \tag{4.55}$$

may integrate all the above cases into a single representation, where u_{true} is the actuator output, u_{cmd} is the output of the controller (which at the same time is an input to the actuator), $\sigma = 1$ and $k = 1$ in the no fault operating regime, $\sigma = 1$ and $\varepsilon < k < 1$ in the case of loss of effectiveness fault and $\sigma = 0$ in other types of fault scenarios. Finally $u_m \leq \bar{u} \leq u_M$ is the position at which the actuator locks in case of float, lock-in-place and hard-over faults. It is assumed that the fault has already been detected and isolated by using nonlinear geometric FDI scheme. For the cases where the FDI algorithm just detect the occurrence of the fault in one of the input channels pairs F_1, F_3 and F_2, F_4, the proposed fault diagnosis algorithm will be applied to each channel in above pairs one by one.

The general idea is to apply some test signals to the system and use estimator to identify the type of fault. The main challenge is to distinguish between the loss of effectiveness fault and other types of fault. This is due to the fact that in the case of float, lock in place and hard over faults, the real value of the output is constant but for loss of effectiveness fault, the actuator output is not necessary constant. Therefore, in order to have the same type of signal as the actuator output, one can choose the piece-wise constant auxiliary signal and in this case, all faults show up as a bias in the actuator and one can model the system as below

$$\dot{x} = f(x) + g(x)(u_a + b_f) \tag{4.56}$$

$$y = Cx \tag{4.57}$$

where b_f denotes the constant bias of the actuator due to the all type of fault and u_a is the auxiliary input signal which is constant and $u_{true} = u_a + b_f$. The problem is now to design an estimator for b_f. According to the type of faults that have been considered here, the following auxiliary signal can be used:

$$u_a = \begin{cases} 0 & t_s \leq t \leq t_s + T \\ a & t_s + T \leq t \leq t_s + 2T \end{cases} \tag{4.58}$$

wheret_s is the time instant that the vehicle switch to the safe mode, T is the required time for the convergence of the estimator. Let $\hat{b}_f$ denotes the estimated value of b_f. It should be noted that it is not always necessary to inject the auxiliary signal with two different levels. When the zero auxiliary signal will be applied to the vehicle, if float or loss of effectiveness fault has happened, then $\hat{b}_f$ will converge to zero value and one cannot distinguish between these type of fault and nonzero auxiliary signal should be applied to identify the fault. However, if LIP or HOV fault has happened in the system, $\hat{b}_f$ will converge to the position of the actuator where it has locked and there is no need to apply nonzero auxiliary signal. In the case of float or loss of effectiveness fault, the nonzero auxiliary signal $u_a = a$ will be applied to the vehicle and one can find the effectiveness of the input channel as follow:

$$k = \frac{a + \hat{b}_f}{a} \times 100 \tag{4.59}$$

Figures 4.125 and 4.126 show the corresponding auxiliary signals a and $\hat{b}_f$ for two different fault scenarios, hard over fault and 80% loss of effectiveness in F_1 actuator, respectively. As it is seen in figure 4.125, there is no need to apply two different level and $\hat{b}_f = 3.5$ determine the type of fault which in this scenario is hard over. However in figure 4.126, applying the zero auxiliary signal results in $\hat{b}_f = 0$ will opt out the possibility of hard over and loss in place fault and the nonzero one can be used to find the percentage of the loss of effectiveness.

4.3.9 Fault recovery for ALTAV

In this section, fault recovery procedures that can be investigated for ALTAV in future is summarized. Since ALTAV is not overactuated system, the control allocation procedures for recovery are not applicable for this system. However one can investigate the controllability of the faulty vehicle for different fault scenario and for each of them, reconfigure the controller in order to achieve the maximum attainable performance.

As an example, consider the float fault in input channel F_1. In this case, state space equation of ALTAV may be rewritten as

$$\begin{aligned} M\ddot{x} &= (F_2 + F_3 + F_4)sin(\gamma) - C_x\dot{x} \\ M\ddot{y} &= (F_2 + F_3 + F_4)sin(\phi) - C_y\dot{y} \\ M\ddot{z} &= -(F_2 + F_3 + F_4)cos(\gamma)cos(\phi) - F_B + Mg - C_z\dot{z} \\ J_\theta\ddot{\theta} &= (-F_2l + F_3l - F_4l)sin(\rho) - C_\theta\dot{\theta} \\ J_\gamma\ddot{\gamma} &= -F_3l - F_BL_Bsin(\gamma) - C_\gamma\dot{\gamma} \\ J_\phi\ddot{\phi} &= -(F_2l - F_4l) - F_BL_Bsin(\phi) - C_\phi\dot{\phi} \end{aligned} \tag{4.60}$$

A recovery controller should be design for this faulty system by considering the subset of states that are controllable and whether the vehicle has full maneuverability in this fault

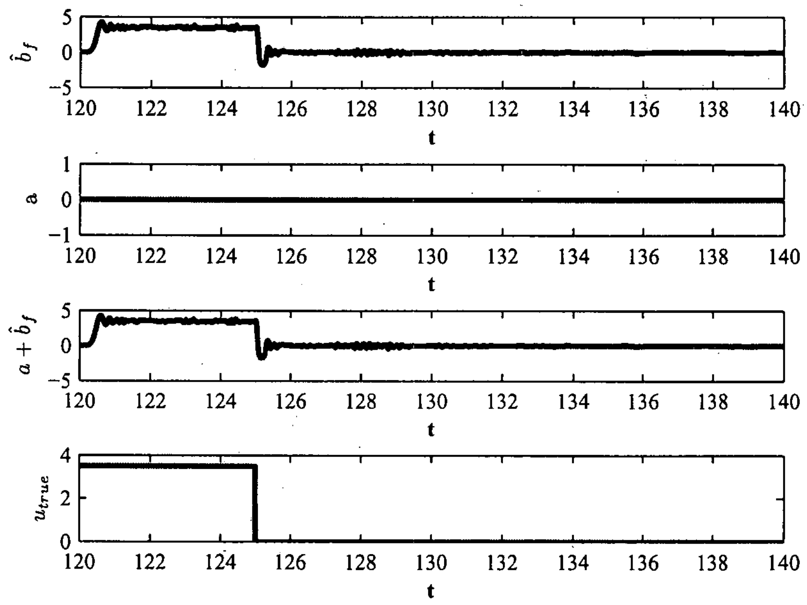


Figure 4.125: Diagnosis signals corresponding to the Hard-over fault in F_1 actuator

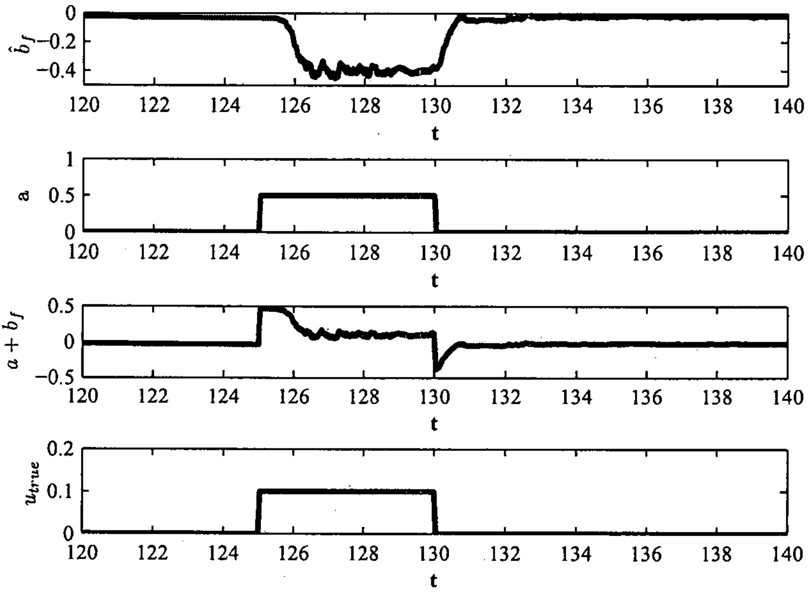


Figure 4.126: Diagnosis signals corresponding to the 80% loss of effectiveness fault in F_1 actuator

scenario. One solution may be turning off the actuator F_3 again and control the vehicle by F_2 and F_4. It should be investigated that whether the vehicle can be controlled in both y and z direction in contrast to the safe mode operation that the vehicle is only controlled in z direction.

Fault Type	Detection Time (sec)	Isolation time (sec)
Float	105.1	107
Hard Over	105.05	105.1
Lock in place	107.1	-
20% LOS	-	-
50% LOS	105.1	-
80% LOS	105.05	111

Table 4.13: Detection time corresponding to fault scenarios of F_1 input channel

As a second example, consider the 50% loss of effectiveness in input channel F_2. In this case the state space equation of ALTAV my be written as

$$\begin{aligned}
M\ddot{x} &= (F_1 + 0.5 * F_2 + F_3 + F_4)sin(\gamma) - C_x\dot{x} \\
M\ddot{y} &= (F_1 + 0.5 * F_2 + F_3 + F_4)sin(\phi) - C_y\dot{y} \\
M\ddot{z} &= -(F1 + 0.5 * F_2 + F_3 + F_4)cos(\gamma)cos(\phi) - F_B + Mg - C_z\dot{z} \\
J_\theta\ddot{\theta} &= (F_1 l - 0.5 * F_2 l + F_3 l - F_4 l)sin(\rho) - C_\theta\dot{\theta} \\
J_\gamma\ddot{\gamma} &= (F1 - F_3)l - F_B L_B sin(\gamma) - C_\gamma\dot{\gamma} \\
J_\phi\ddot{\phi} &= -(0.5 * F_2 - F_4)l - F_B L_B sin(\phi) - C_\phi\dot{\phi}
\end{aligned} \tag{4.61}$$

As it shown in figure 4.64, the normal control can control the vehicle in x and z direction completely but there exist a tracking error in y direction motion. Therefore, the vehicle still has maneuverability in x and z direction which it also should be investigated analytically. Also it should be investigate what is the maximum performance attainable in y direction.

As a summary, the behavior of vehicle for all type of faults should be investigated and control recovery configuration for each fault scenario should be designed.

4.3.10 Timing issue in FDIR

In this section, timing issue of the proposed FDIR is considered. As it is mentioned before, the first step in FDIR is fault detection and isolation. Different criteria can be chosen for fault decision making. As an example, we can set a fault alarm when the flag corresponding to the faulty channel is one for more than 5 second. This scheme may decrease the false alarm. Following table shows the alarm time of different fault scenarios in input channels for this case. All faults are injected in $t = 100$ second.

Other decision making rule such as stochastic criteria for detection of abrupt change can be used which guarantee the minimum false alarm.

After the fault is detected and isolated by the FDI algorithm, the controller will switch to the safe mode and fault diagnosis algorithm is activated. Fault diagnosis require 10 second for identifying the fault in the vehicle. This interval can be optimized to find the minimum time for fault diagnosis algorithm. At the last step, the controller should be reconfigured according to the type of fault. The most important time is the detection fault since during this interval the vehicle is faulty but the normal operation controller is applied to the vehicle and this may cause unstability and damage to the vehicle or other vehicle in the neighborhood

Fault Type	Detection Time (sec)	Isolation time (sec)
Float	105.05	109.5
Hard Over	108	118
Lock in place	107.0	-
20% LOS	-	-
50% LOS	105.1	-
80% LOS	105.05	-

Table 4.14: Detection time corresponding to fault scenarios of F_2 input channel

Fault Type	Detection Time (sec)	Isolation time (sec)
Float	105	107
Hard Over	108	109
Lock in place	107	-
20% LOS	-	-
50% LOS	105.1	-
80% LOS	105.05	-

Table 4.15: Detection time corresponding to fault scenarios of F_3 input channel

Fault Type	Detection Time (sec)	Isolation time (sec)
Float	105.1	111
Hard Over	105.05	105.1
Lock in place	106.1	-
20% LOS	-	-
50% LOS	113	-
80% LOS	105.05	-

Table 4.16: Detection time corresponding to fault scenarios of F_4 input channel

of faulty vehicle. Therefore detection decision making should be optimized and the maximum detection time for all type fault (worse case) must be considered in the design of controller in the normal operation. These are the issue that should be investigated in future.

4.4 Aerosonde FDI

Aerosonde [53, 4] is a small autonomous airplane designed for long-range weather data acquisition. It was developed especially for reconnaissance over oceanic and remot areas, and in harsh conditions. Its low-cost design and operational flexibility will promote its usage for a wide range of remote-sensing application. In this section, the nonlinear geometric FDI will be applied to the lateral dynamics of the Aerosonde.

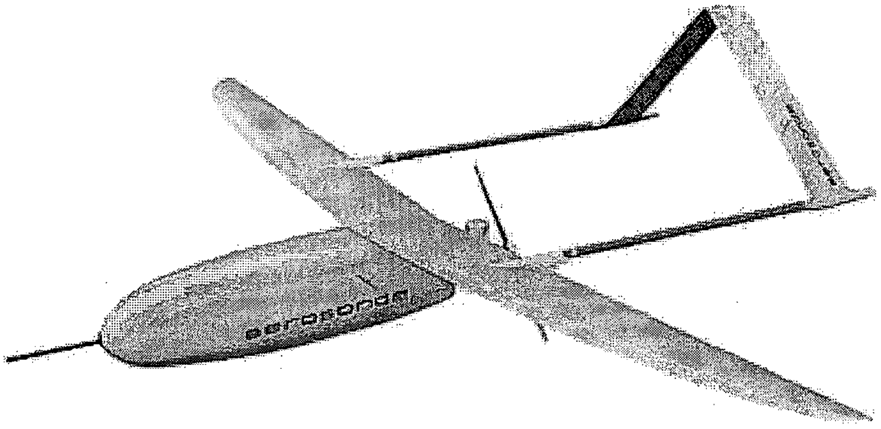


Figure 4.127: The Aerosonde [4]

4.4.1 Aerosonde Model

Aerosonde's dynamics and kinematics can be written by the following equations []:

Force Equations

$$\begin{aligned} m_v\dot{V_T} &= F_T cos(\alpha+\alpha_T)cos\beta - D + m_v g_1 && (4.62)\\ m_v\dot{\beta}V_T &= -F_T cos(\alpha+\alpha_T)sin\beta + C + m_v g_2 - m_v V_T r_S && (4.63)\\ m_v\dot{\alpha}V_T cos\beta &= -F_T sin(\alpha+\alpha_T) - L + m_v g_3 + m_v V_T(qcos\beta - p_S sin\beta) && (4.64) \end{aligned}$$

where

$$\begin{aligned} g_1 &= -g_D cos\beta sin(\theta-\alpha) && (4.65)\\ g_2 &= g_D sin\beta sin(\theta-\alpha) && (4.66)\\ g_3 &= g_D cos(\theta-\alpha) && (4.67) \end{aligned}$$

Moment Equations

$$\begin{aligned} \Gamma\dot{p} &= J_{xz}[J_x - J_y + J_z]pq + [J_z(J_y - J_z) - J_{xz}^2]qr + J_z l + J_{xz}n && (4.68)\\ J_y\dot{q} &= (J_z - J_x)pr - J_{xz}(p^2 - r^2) + m && (4.69)\\ \Gamma\dot{r} &= [(J_x - J_y)J_x + J_{xz}^2]pq - J_{xz}[J_x - J_y + J_z]qr + J_{xz}l + J_x n && (4.70) \end{aligned}$$

where $\Gamma = J_x J_z - J_{xz}^2$.

Kinematic Equations

$$\begin{aligned} \dot{\phi} &= p + tan\theta(qsin\phi + rcos\phi) && (4.71)\\ \dot{\theta} &= qcos\phi - rsin\phi && (4.72)\\ \dot{\psi} &= (qsin\phi + rcos\phi)/cos\theta && (4.73) \end{aligned}$$

Navigation Equations

$$\begin{aligned} \dot{p_N} &= Uc\theta c\psi + V(-c\phi s\psi + s\phi s\theta s\psi) + W(s\phi s\psi + c\phi s\theta c\psi) && (4.74)\\ \dot{p_E} &= Uc\theta s\psi + V(c\phi c\psi + s\phi s\theta s\psi) + W(-s\phi c\psi + c\phi s\theta s\psi) && (4.75)\\ \dot{h} &= Us\theta - Vs\phi c\theta - Wc\phi c\theta && (4.76) \end{aligned}$$

We will focus on force equations and moment equations in order to design FDI techniques. Aerodynamic forces and moments are obtained based on following equations.

Aerodynamic forces:

$$D = \frac{1}{2}\rho V_T^2 S C_D \tag{4.77}$$

$$C = \frac{1}{2}\rho V_T^2 S C_Y \tag{4.78}$$

$$L = \frac{1}{2}\rho V_T^2 S C_L \tag{4.79}$$

Aerodynamic moments:

$$l = \frac{1}{2}\rho V_T^2 S b C_l \tag{4.80}$$

$$m = \frac{1}{2}\rho V_T^2 S c C_m \tag{4.81}$$

$$n = \frac{1}{2}\rho V_T^2 S b C_n \tag{4.82}$$

The coefficient terms in the above equations can be modeled as follows:

Lift coefficient terms:

$$C_L = C_{L0} + C_L^{\alpha} \cdot \alpha + C_L^{\delta_f} \cdot \delta_f + C_L^{\delta_e} \cdot \delta_e + \frac{c}{2V_T}(C_L^{\dot{\alpha}} \cdot \dot{\alpha} + C_L^{q} \cdot q) + C_L^{M} \cdot M \tag{4.83}$$

Drag coefficient terms:

$$C_D = C_{D0} + \frac{(C_L - C_{L0})^2}{\pi e AR} + C_D^{\delta_f} \cdot \delta_f + C_D^{\delta_e} \cdot \delta_e + C_D^{\delta_a} \cdot \delta_a + C_D^{\delta_r} \cdot \delta_r + C_D^{M} \cdot M \tag{4.84}$$

Side force coefficient terms:

$$C_Y = C_Y^{\beta} \cdot \beta + C_Y^{\delta_a} \cdot \delta_a + C_Y^{\delta_r} \cdot \delta_r + \frac{b}{2V_a}(C_Y^{p} \cdot p + C_Y^{r} \cdot r) \tag{4.85}$$

Pitch moment coefficient terms:

$$C_m = C_{m0} + C_m^{\alpha} \cdot \alpha + C_m^{\delta_f} \cdot \delta_f + C_m^{\delta_e} \cdot \delta_e + \frac{c}{2V_T}(C_m^{\dot{\alpha}} \cdot \dot{\alpha} + C_m^{q} \cdot q) + C_m^{M} \cdot M \tag{4.86}$$

Roll moment coefficient terms:

$$C_l = C_l^{\beta} \cdot \beta + C_l^{\delta_a} \cdot \delta_a + C_l^{\delta_r} \cdot \delta_r + \frac{b}{2V_T}(C_l^{p} \cdot p + C_l^{r} \cdot r) \tag{4.87}$$

Yaw moment coefficient terms:

$$C_n = C_n^{\beta} \cdot \beta + C_n^{\delta_a} \cdot \delta_a + C_n^{\delta_r} \cdot \delta_r + \frac{b}{2V_T}(C_n^{p} \cdot p + C_n^{r} \cdot r) \tag{4.88}$$

Therefore, considering these coefficients, force equations in the wind axis can be rewritten as:

$$\begin{aligned} m_v\dot{V_T} = F_T cos(\alpha + \alpha_T) cos\beta - \frac{1}{2}\rho V_T^2 S\Bigg[C_{D0} + \frac{(C_L - C_{L0}^2)}{\pi e AR} + C_D^{\delta_f} \cdot \delta_f \\ + C_D^{\delta_e} \cdot \delta_e + C_D^{\delta_a} \cdot \delta_a + C_D^{\delta_r} \cdot \delta_r + C_D^M \cdot M\Bigg] + m_v g_1 \end{aligned} \tag{4.89}$$

$$\begin{aligned} m_v\dot{\beta} V_T = -F_T cos(\alpha + \alpha_T) sin\beta + \frac{1}{2}\rho V_T^2 S\Bigg[C_Y^{\beta} \cdot \beta + C_Y^{\delta_a} \cdot \delta_a + C_Y^{\delta_r} \cdot \delta_r \\ + \frac{b}{2V_T}(C_Y^p \cdot p + C_Y^r \cdot r)\Bigg] + m_v g_2 - m_v V_T r_S \end{aligned} \tag{4.90}$$

$$\begin{aligned} m_v\dot{\alpha} V_T cos\beta = -F_T sin(\alpha + \alpha_T) - \frac{1}{2}\rho V_T^2 S\Bigg[C_{L0} + C_L^{\alpha} + C_L^{\delta_f} \cdot \delta_f + C_L^{\delta_e} \cdot \delta_e \\ + \frac{c}{2V_T}(C_L^{\dot{\alpha}} \cdot \dot{\alpha} + C_L^q \cdot q) + C_L^M \cdot M\Bigg] + m_v g_3 + m_v V_T(q cos\beta - p_S sin\beta) \end{aligned} \tag{4.91}$$

4.4.2 Parameter List

Aerosonde parameter's values are summarized in following tables [54].

Notation	**Description**	**Values**
C_{L0}	zero-alpha lift	0.23
$C_L^{\dot{\alpha}}$	alpha derivative	5.6106
$C_L^{\delta_f}$	lift control (flap) derivative	0.74
$C_L^{\delta_e}$	pitch control (elevator) derivative	0.13
$C_L^{\dot{\alpha}}$	alpha-dot derivative	1.9724
C_L^q	pitch rate derivative	7.9543
C_L^M	Mach number derivative	0

Table 4.17: Lift Coefficients

Notation	**Description**	**Values**
C_{D0}	minimum drag	0.0434
$C_D^{\delta_f}$	lift control (flap) derivative	0.1467
$C_D^{\delta_e}$	pitch control (elevator) derivative	0.0135
$C_D^{\delta_a}$	roll control(aileron) derivative	0.0303
$C_D^{\delta_r}$	yaw control (rudder) derivative	0.0303
C_D^M	Mach number derivative	0

Table 4.18: Drag Coefficients

Notation	Description	Values
C_Y^{β}	sideslip derivative	-0.83
$C_Y^{\delta_a}$	roll control (aileron) derivative	-0.075
$C_Y^{\delta_r}$	yaw control (rudder) derivative	0.1914
C_Y^{p}	roll rate derivative	0
C_Y^{r}	yaw rate derivative	0

Table 4.19: Side Force Coefficients

Notation	Description	Values
C_{m0}	zero-alpha pitch	0.135
C_m^{α}	alpha derivative	-2.7397
$C_m^{\delta_f}$	lift control (flap) derivative	0.0467
$C_m^{\delta_e}$	pitch control (elevator) derivative	-0.9918
$C_m^{\dot{\alpha}}$	alpha-dot derivative	-10.3796
C_m^{q}	pitch rate derivative	-38.2067
C_m^{M}	Mach number derivative	0

Table 4.20: Pitch Moment Coefficients

Notation	Description	Values
C_l^{β}	sideslip derivative	-0.13
$C_l^{\delta_a}$	roll control(aileron) derivative	-0.1695
$C_l^{\delta_r}$	yaw control (rudder) derivative	0.0024
C_l^{p}	roll rate derivative	-0.5051
C_l^{r}	yaw rate derivative	0.2519

Table 4.21: Roll Moment Coefficients

Notation	Description	Values
C_n^{β}	sideslip derivative	-0.0726
$C_n^{\delta_a}$	roll control (aileron) derivative	0.0108
$C_n^{\delta_r}$	yaw control (rudder) derivative	-0.0693
C_n^{p}	roll rate derivative	-0.069
C_n^{r}	yaw rate derivative	-0.0946

Table 4.22: Yaw Moment Coefficients

Notation	Description	Values
c	mean aerodynamic chord	0.189941 m
b	wind span	2.8956 m
S	wing area	0.55 m^2

Table 4.23: Aerodynamic Parameters

	notation	zero-fuel	full fuel tank
mass	m_v	8.5 kg	13.5 kg
moments of inertia	J_x	0.7795 $kg \cdot m^2$	0.8244 $kg \cdot m^2$
	J_y	1.122 $kg \cdot m^2$	1.135 $kg \cdot m^2$
	J_z	1.752 $kg \cdot m^2$	1.759 $kg \cdot m^2$
	J_{xz}	0.1211 $kg \cdot m^2$	0.1204 $kg \cdot m^2$
CG location	x	0.156 m	0.159 m
	y	0 m	0 m
	z	0.079 m	0.090 m

Table 4.24: Inertia

4.4.3 Lateral motion

For lateral motion, we consider that angle of attack α , pitch angle and pitch rate are zero. Thus, the lateral dynamics of aerosonde can be written as follows

$$\begin{aligned} m_v\dot{V_T} &= F_T\cos\beta - \frac{1}{2}\rho V_T^2 S\left[C_{D0} + C_D^{\delta_a}\cdot\delta_a + C_D^{\delta_r}\cdot\delta_r\right] && (4.92)\\ m_v\dot{\beta} V_T &= -F_T sin\beta + \frac{1}{2}\rho V_T^2 S\left[C_Y^{\beta}\cdot\beta + C_Y^{\delta_a}\cdot\delta_a + C_Y^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_Y^p\cdot p + C_Y^r\cdot r)\right] - m_v V_T r \\ \Gamma\dot{p} &= J_z\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_l^{\beta}\cdot\beta + C_l^{\delta_a}\cdot\delta_a + C_l^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_l^p\cdot p + C_l^r\cdot r)\right] \\ &\quad + J_{xz}\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_n^{\beta}\cdot\beta + C_n^{\delta_a}\cdot\delta_a + C_n^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_n^p\cdot p + C_n^r\cdot r)\right] && (4.93)\\ \Gamma\dot{r} &= J_{xz}\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_l^{\beta}\cdot\beta + C_l^{\delta_a}\cdot\delta_a + C_l^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_l^p\cdot p + C_l^r\cdot r)\right] \\ &\quad + J_z\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_n^{\beta}\cdot\beta + C_n^{\delta_a}\cdot\delta_a + C_n^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_n^p\cdot p + C_n^r\cdot r)\right] \\ & && (4.94) \end{aligned}$$

where the inputs are F_T, δ_a and δ_r. Consider $x = [V_T \quad \beta \quad p \quad r]$, the above equations can be written in compact form

$$\dot{x} = f(x) + g_1(x)F_T + g_2(x)\delta_a + g_3(x)\delta_r \qquad (4.95)$$

where

$$f(x) = \begin{bmatrix} -\frac{\rho x_1^2 S C_{D0}}{2m_v} \\ -\frac{\rho x_1 S C_Y^{\beta}}{2m_v}\cdot x_2 - x_4 \\ J_z\cdot\frac{1}{2\Gamma}\rho x_1^2 Sb\left[C_l^{\beta}\cdot x_2 + \frac{b}{2x_1}(C_l^p\cdot x_3 + C_l^r\cdot x_4)\right] \\ +J_{xz}\cdot\frac{1}{2\Gamma}\rho x_1^2 Sb\left[C_n^{\beta}\cdot x_2 + \frac{b}{2x_1}(C_n^p\cdot x_3 + C_n^r\cdot x_4)\right] \\ \\ J_{xz}\cdot\frac{1}{2\Gamma}\rho x_1^2 Sb\left[C_l^{\beta}\cdot x_2 + \frac{b}{2x_1}(C_l^p\cdot x_3 + C_l^r\cdot x_4)\right] \\ +J_z\cdot\frac{1}{2\Gamma}\rho x_1^2 Sb\left[C_n^{\beta}\cdot x_2 + \frac{b}{2x_1}(C_n^p\cdot x_3 + C_n^r\cdot x_4)\right] \\ \end{bmatrix}$$

$$g_1(x) = \begin{bmatrix} \frac{\cos x_2}{m_v} \\ -\frac{\sin x_2}{m_v x_1} \\ 0 \\ 0 \end{bmatrix},$$

$$g_2(x) = \begin{bmatrix} -\frac{\rho x_1^2 S}{2m_v} C_D^{\delta_a} \\ \frac{\rho x_1 S}{2m_v} C_Y^{\delta_a} \\ \frac{\rho x_1^2 Sb}{2\Gamma}\left[J_z C_l^{\delta_a} + J_{xz} C_n^{\delta_a}\right] \\ \frac{\rho x_1^2 Sb}{2\Gamma}\left[J_{xz} C_l^{\delta_a} + J_z C_n^{\delta_a}\right] \end{bmatrix}$$

$$g_3(x) = \begin{bmatrix} -\frac{\rho x_1^2 S}{2m_v} C_D^{\delta_r} \\ \frac{\rho x_1 S}{2m_v} C_Y^{\delta_r} \\ \frac{\rho x_1^2 Sb}{2\Gamma}\left[J_z C_l^{\delta_r} + J_{xz} C_n^{\delta_r}\right] \\ \frac{\rho x_1^2 Sb}{2\Gamma}\left[J_{xz} C_l^{\delta_r} + J_z C_n^{\delta_r}\right] \end{bmatrix}$$

4.4.4 Aerosonde Control System

In this section, a PID control will be applied to lateral motion of Aerosonde. More specifically, aileron, rudder and throttle will be used to control roll angle, yaw angle and airspeed, respectively.

It is important to note that the airplane thrust F_T in (6.8) – (6.8) is one of the inputs. The airplane thrust is provided by propulsion system, which is affected by throttle, the atmospheric pressure, the atmospheric temperature, the air density, etc. The input of the propulsion system in our disposal is throttle which ranges from 0 to 1. Consequently, in the following simulation, the throttle is used to control airspeed.

The performance of the PID control of the Aerosonde lateral motion is shown by Figures 4.128 – 4.133. Figures 4.128 – 4.130 depict the performance of yaw, airspeed and roll of Aerosonde. Yaw and airspeed are expected to follow the steps shown by red lines in the figures, and roll is expected to stay zero during the flight. One can observe that the designed PID control can make the UAV follow the desired references, and the input responses of the UAV are shown by Figures 4.131 – 4.133.

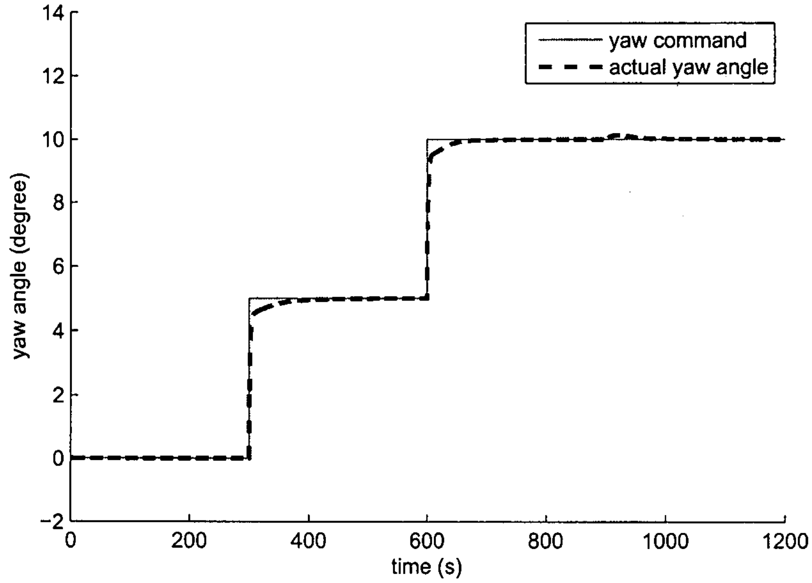


Figure 4.128: Performance of yaw angle

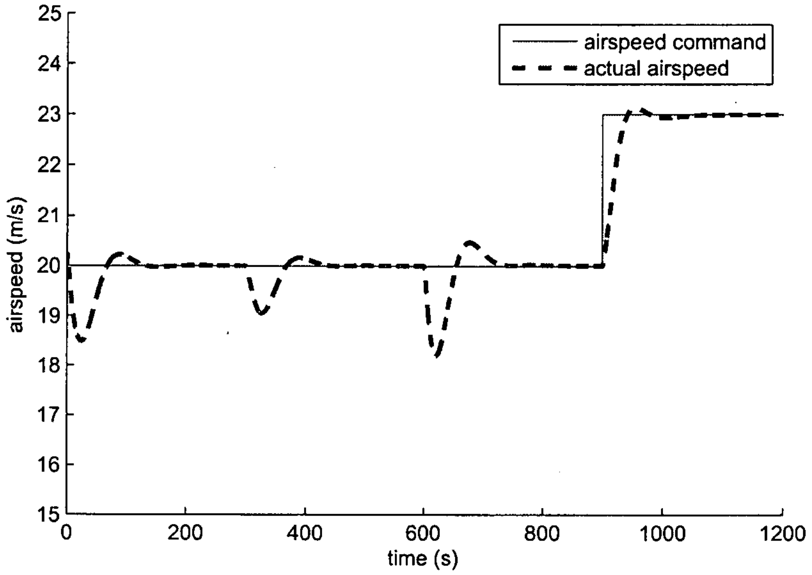


Figure 4.129: Performance of airspeed of Aerosonde

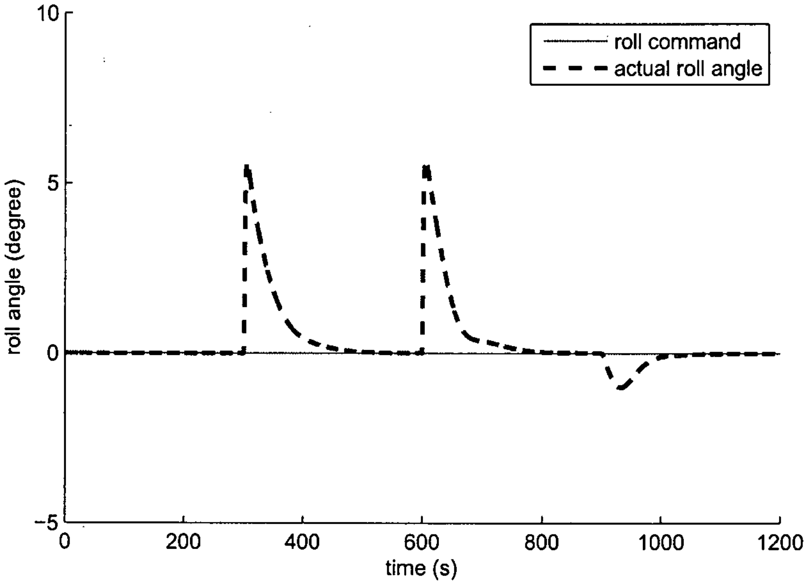


Figure 4.130: Performance of roll angle

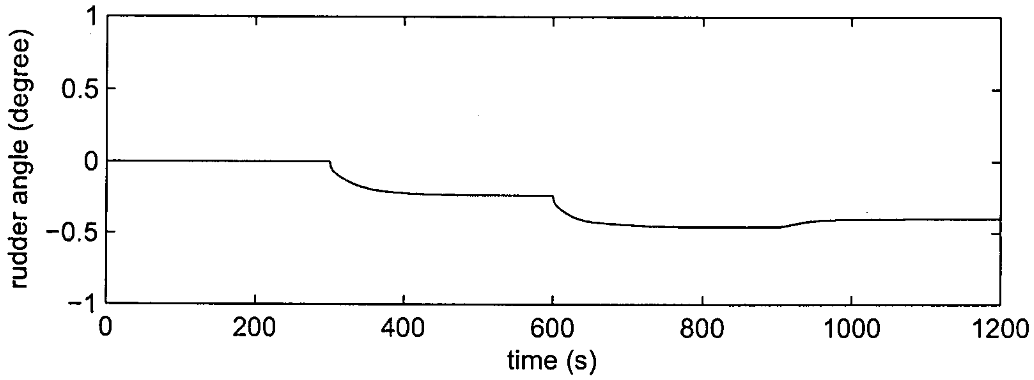


Figure 4.131: Input response of rudder

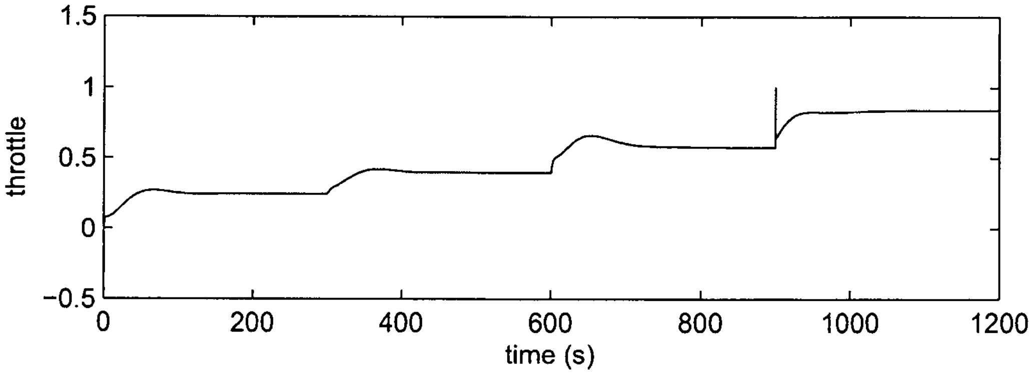


Figure 4.132: Input response of throttle

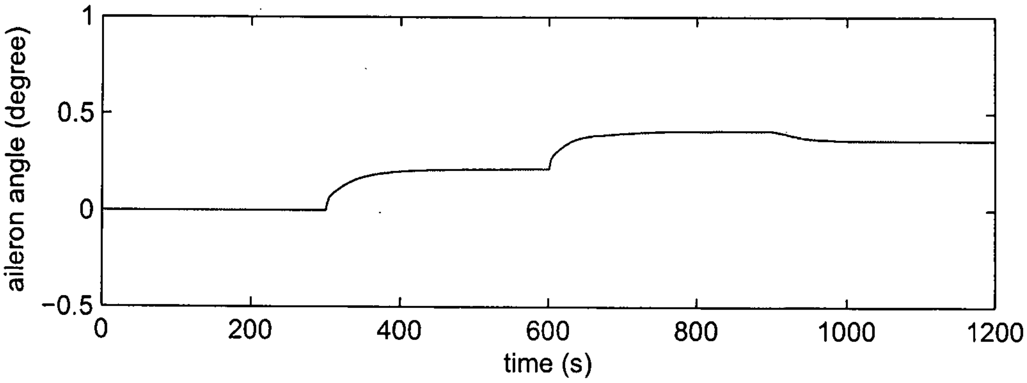


Figure 4.133: Input response of aileron

4.4.5 Aerosonde Fault detection and Isolation

In this section the fault detection and isolation problem for lateral dynamics of aerosonde (Equation (6.8)) is investigated. The nonlinear geometric FDI based detection filters are designed for different input channels. Three input channels *aileron* δ_a *, rudder* δ_r *and throttle* F_T are used for controlling the lateral motion of aerosonde. Due to dynamics of aerosonde, it is not possible to find a residual that is just affected by one channel and is decoupled from other channels. Therefore, a set of three detection filters are designed such that each residual is decoupled from one input channel and is affected by two other channels. By using the nonlinear geometric FDI approach following new set of states can be found:

$$z_1 = log(x_1) + x_2 \frac{C_D^{\delta_a}}{C_Y^{\delta_a}} \tag{4.96}$$

$$z_2 = log(x_1) + x_2 \frac{C_D^{\delta_r}}{C_Y^{\delta_r}} \tag{4.97}$$

$$z_3 = p \tag{4.98}$$

$$z_4 = r \tag{4.99}$$

where the inverse mapping between the new states and the old states are

$$x_1 = exp(\frac{z_1 - \frac{C_D^{\delta_a} C_Y^{\delta_r}}{C_Y^{\delta_a} C_D^{\delta_r}} z_2}{1 - \frac{C_D^{\delta_a} C_Y^{\delta_r}}{C_Y^{\delta_a} C_D^{\delta_r}}}) \tag{4.100}$$

$$x_2 = \frac{z_1 - z_2}{\frac{C_D^{\delta_a}}{C_Y^{\delta_a}} - \frac{C_D^{\delta_r}}{C_Y^{\delta_r}}} \tag{4.101}$$

$$x_3 = z_3 \tag{4.102}$$

$$x_4 = z_4 \tag{4.103}$$

Let $C_a = \frac{C_D^{\delta_a}}{C_Y^{\delta_a}}$ and $C_r = \frac{C_D^{\delta_r}}{C_Y^{\delta_r}}$ then, the state space equation for new set of states are:

$$
\begin{aligned}
\dot{z}_1 = {} & \frac{1}{m_v exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}[F_T \cos(\frac{z_1 - z_2}{C_a - C_r}) - F_T sin(\frac{z_1 - z_2}{C_a - C_r})] - \frac{C_D^{\delta_a}}{C_Y^{\delta_a}} z_4 \\
& + \frac{1}{2m_v} \rho exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a}) S \\
& + \left[\frac{C_D^{\delta_a}}{C_Y^{\delta_a}} (C_Y^{\beta} \cdot \beta + C_Y^{\delta_r} \cdot \delta_r + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_Y^p \cdot z_3 + C_Y^r \cdot z_4)) - C_{D0} + C_D^{\delta_r} \cdot \delta_r \right] \\
\dot{z}_2 = {} & \frac{1}{m_v exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}[F_T \cos(\frac{z_1 - z_2}{C_a - C_r}) - F_T sin(\frac{z_1 - z_2}{C_a - C_r})] - \frac{C_D^{\delta_r}}{C_Y^{\delta_r}} z_4 \\
& + \frac{1}{2m_v} \rho exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a}) S \\
& + \left[\frac{C_D^{\delta_r}}{C_Y^{\delta_r}} (C_Y^{\beta} \cdot \beta + C_Y^{\delta_a} \cdot \delta_a + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_Y^p \cdot z_3 + C_Y^r \cdot z_4)) - C_{D0} + C_D^{\delta_a} \cdot \delta_a \right] \\
\dot{z}_3 = {} & J_z \cdot \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[C_l^{\beta} \cdot (\frac{z_1 - z_2}{C_a - C_r}) + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_l^p \cdot z_3 + C_l^r \cdot z_4) \right] \\
& + J_{xz} \cdot \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[C_n^{\beta} \cdot (\frac{z_1 - z_2}{C_a - C_r}) + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_n^p \cdot z_3 + C_n^r \cdot z_4) \right] \\
& + \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[J_z C_l^{\delta_a} + J_{xz} C_n^{\delta_a} \right] \cdot \delta_a \\
& + \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[J_z C_l^{\delta_r} + J_{xz} C_n^{\delta_r} \right] \cdot \delta_r \\
\dot{z}_4 = {} & J_{xz} \cdot \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[C_l^{\beta} \cdot \beta + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_l^p \cdot p + C_l^r \cdot r) \right] \\
& + J_z \cdot \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[C_n^{\beta} \cdot \beta + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_n^p \cdot p + C_n^r \cdot r) \right] \\
& + \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[J_{xz} C_l^{\delta_a} + J_z C_n^{\delta_a} \right] \cdot \delta_a \\
& + \frac{1}{2\Gamma} \rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a}) Sb \left[J_{xz} C_l^{\delta_r} + J_z C_n^{\delta_r} \right] \cdot \delta_r
\end{aligned}
$$

As it is seen from above equation z_1 is affected by F_T and δ_r, z_2 is just affected by F_T and δ_a and z_3 is just affected by δ_a and δ_r. Therefore one need to design an observer for z_1,z_2 and z_3. For designing the observer for z_i, all other states $z_j, j \neq i$ are considered as the independent inputs. Therefore following three high gain observers may be designed:

$$\begin{aligned}
\dot{\hat{z}}_1 = & \frac{1}{m_v exp(\frac{C_r\hat{z}_1 - C_a z_2}{C_r - C_a})}[F_T\cos(\frac{\hat{z}_1 - z_2}{C_a - C_r}) - F_T sin(\frac{\hat{z}_1 - z_2}{C_a - C_r})] - \frac{C_D^{\delta_a}}{C_Y^{\delta_a}} z_4 \\
& + \frac{1}{2m_v}\rho exp(\frac{C_r\hat{z}_1 - C_a z_2}{C_r - C_a})S \\
& + \left[\frac{C_D^{\delta_a}}{C_Y^{\delta_a}}(C_Y^{\beta}\cdot\beta + C_Y^{\delta_r}\cdot\delta_r + \frac{b}{2exp(\frac{C_r\hat{z}_1 - C_a z_2}{C_r - C_a})}(C_Y^{p}\cdot z_3 + C_Y^{r}\cdot z_4)) - C_{D0} + C_D^{\delta_r}\cdot\delta_r\right] \\
\dot{\hat{z}}_2 = & \frac{1}{m_v exp(\frac{C_r z_1 - C_a \hat{z}_2}{C_r - C_a})}[F_T\cos(\frac{z_1 - \hat{z}_2}{C_a - C_r}) - F_T sin(\frac{z_1 - \hat{z}_2}{C_a - C_r})] - \frac{C_D^{\delta_r}}{C_Y^{\delta_r}} z_4 \\
& + \frac{1}{2m_v}\rho exp(\frac{C_r z_1 - C_a\hat{z}_2}{C_r - C_a})S \\
& + \left[\frac{C_D^{\delta_r}}{C_Y^{\delta_r}}(C_Y^{\beta}\cdot\beta + C_Y^{\delta_a}\cdot\delta_a + \frac{b}{2exp(\frac{C_r z_1 - C_a\hat{z}_2}{C_r - C_a})}(C_Y^{p}\cdot z_3 + C_Y^{r}\cdot z_4)) - C_{D0} + C_D^{\delta_a}\cdot\delta_a\right] \\
\dot{\hat{z}}_3 = & J_z\cdot\frac{1}{2\Gamma}\rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a})Sb\left[C_l^{\beta}\cdot(\frac{z_1 - z_2}{C_a - C_r}) + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_l^{p}\cdot\hat{z}_3 + C_l^{r}\cdot z_4)\right] \\
& + J_{xz}\cdot\frac{1}{2\Gamma}\rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a})Sb\left[C_n^{\beta}\cdot(\frac{z_1 - z_2}{C_a - C_r}) + \frac{b}{2exp(\frac{C_r z_1 - C_a z_2}{C_r - C_a})}(C_n^{p}\cdot\hat{z}_3 + C_n^{r}\cdot z_4)\right] \\
& + \frac{1}{2\Gamma}\rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a})Sb\left[J_z C_l^{\delta_a} + J_{xz}C_n^{\delta_a}\right]\cdot\delta_a \\
& + \frac{1}{2\Gamma}\rho exp(2\frac{C_r z_1 - C_a z_2}{C_r - C_a})Sb\left[J_z C_l^{\delta_r} + J_{xz}C_n^{\delta_r}\right]\cdot\delta_r
\end{aligned}$$

and the residuals can be defined as

$$r_a = z_1 - \hat{z}_1 \tag{4.104}$$

$$r_r = z_2 - \hat{z}_2 \tag{4.105}$$

$$r_F = z_3 - \hat{z}_3 \tag{4.106}$$

Following table will be used for fault detection and isolation decision making:

4.4.6 Simulation Results

In this section, simulation results of our proposed nonlinear FDI scheme when applied to the lateral motion of aerosonde model will be presented. Different actuator faults are considered in all input channels. We have considered a measurements noise v as uniform random noise. Figure 4.134 shows the desired and actual aerosonde lateral states in the normal or healthy operation and figure 4.135 shows the corresponding residuals in this mode. By considering the maximum value of residuals in the normal mode and some safety factor, the threshold value of 0.02 was considered for all residuals.

Threshold exceeds	Threshold does not exceed	detected fault
r_a	r_r, r_F	rudder or engine
r_r	r_a, r_F	aileron or engine
r_F	r_a, r_r	aileron or rudder
r_r, r_F	r_a	aileron
r_a, r_F	r_r	rudder
r_a, r_a	r_F	engine
r_a, r_a, r_F	-	Concurrent faults in two channels or more

Table 4.25: Detection and isolation of Faults in aerosonde

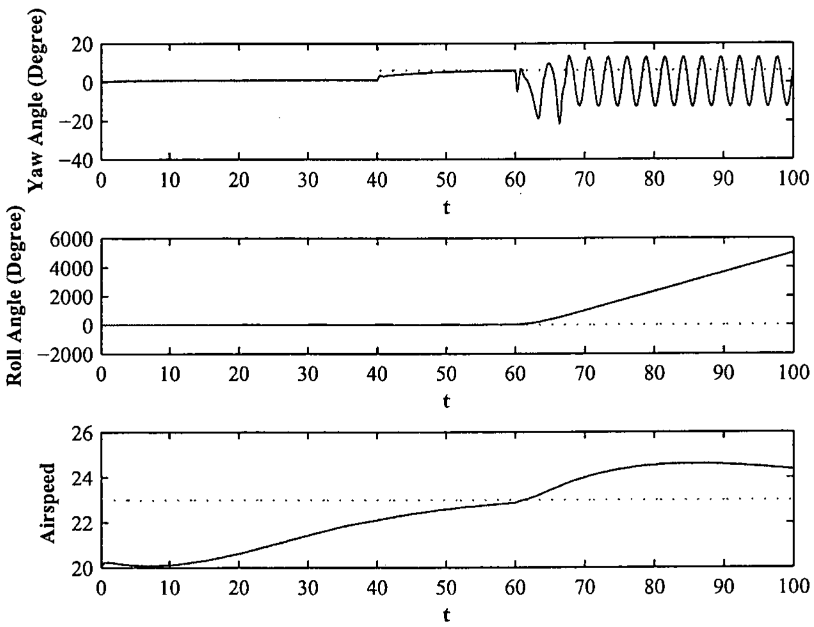


Figure 4.134: Aerosonde states corresponding to a normal operation (healthy mode) of aerosonde

Figure 4.136 shows the residuals corresponding to a float fault in the aileron. As shown in this figure the residuals r_r and r_F exceed their thresholds while residual r_a remains below the threshold. According to table 6.2 one can detect the occurrence of the fault in the aileron. Figure 4.137 demonstrates the states of aerosonde in this faulty scenario which implies that the aerosonde becomes unstable due to this fault. Figures 4.138 and 4.140 show the residuals corresponding to a hard over and lock in place fault in the aileron respectively. According to these figures and table 6.2 , one can easily detect and isolate this fault. As shown in figures 4.139 and 4.141 , the aerosonde becomes unstable in both scenarios. Figure 4.142 depicts the residuals corresponding to a 20% loss of effectiveness fault in the aileron. According to this figure one can only detect the fault but cannot isolate between aileron and rudder. Moreover, as shown in figure 4.143, the controller compensates this fault and there exist just some transient effect on the aerosonde states. Figures 4.144 and 4.146 show the residuals

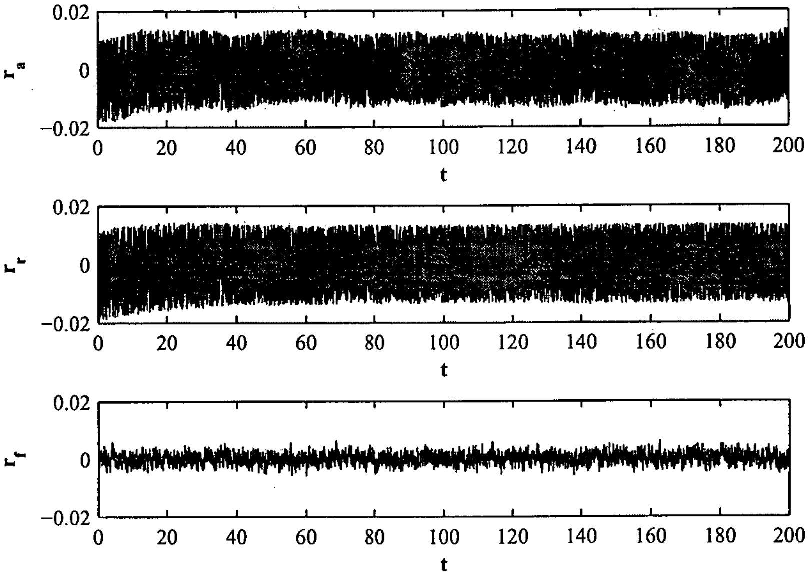


Figure 4.135: Residuals corresponding to a normal operation (healthy mode) of aerosonde

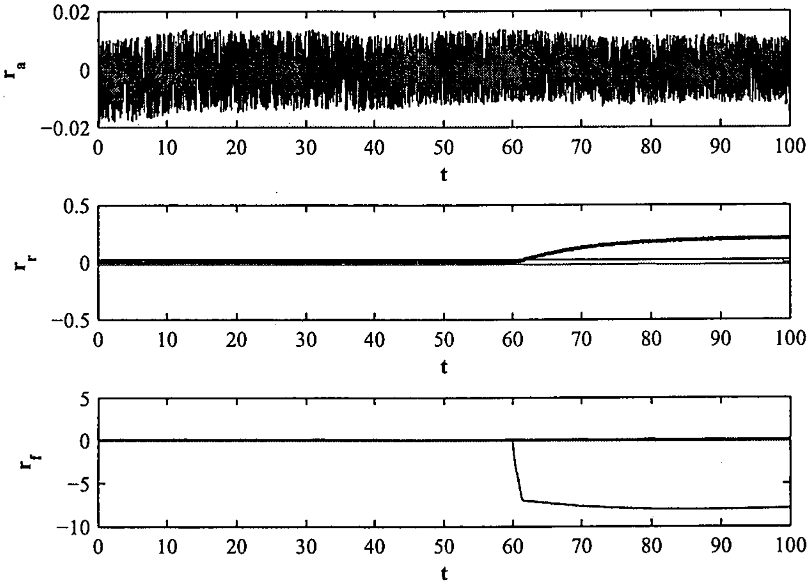


Figure 4.136: Residuals corresponding to a float fault in the aileron

corresponding to the 50% and 80% loss of effectiveness in the aileron respectively. In these two scenarios also, one can detect and isolate the fault in the aileron. Figures 4.145 and 4.147 depict the states of the aerosonde for 50% and 80% loss of effectiveness in the aileron respectively. According to these figures, the controller can recover from the 50% loss of effectiveness fault, while the 80% loss of effectiveness destabilize the aerosonde.

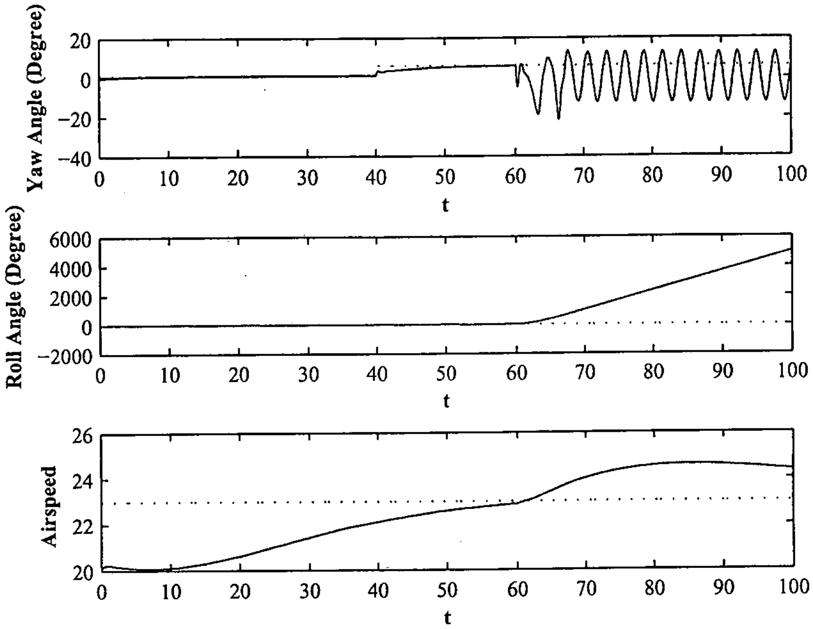


Figure 4.137: Aerosonde states corresponding to a float fault in the aileron

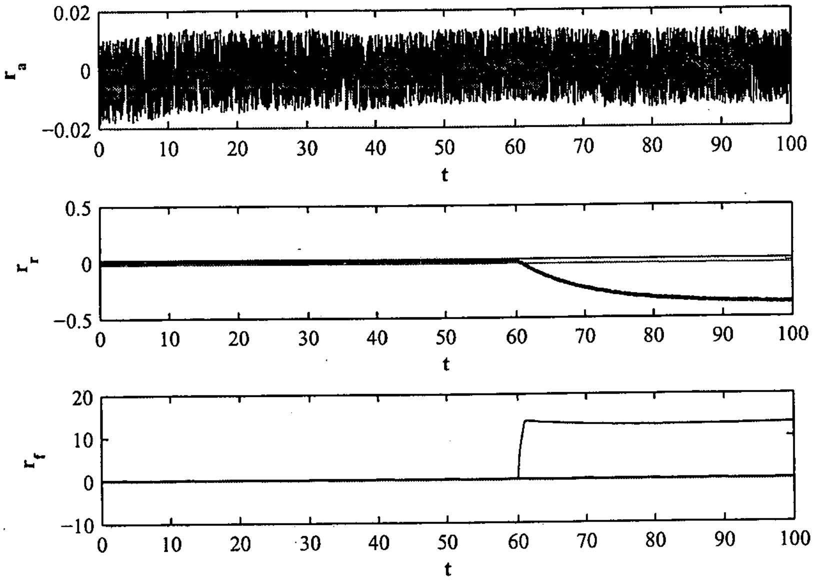


Figure 4.138: Residuals corresponding to a Hard over fault in the aileron

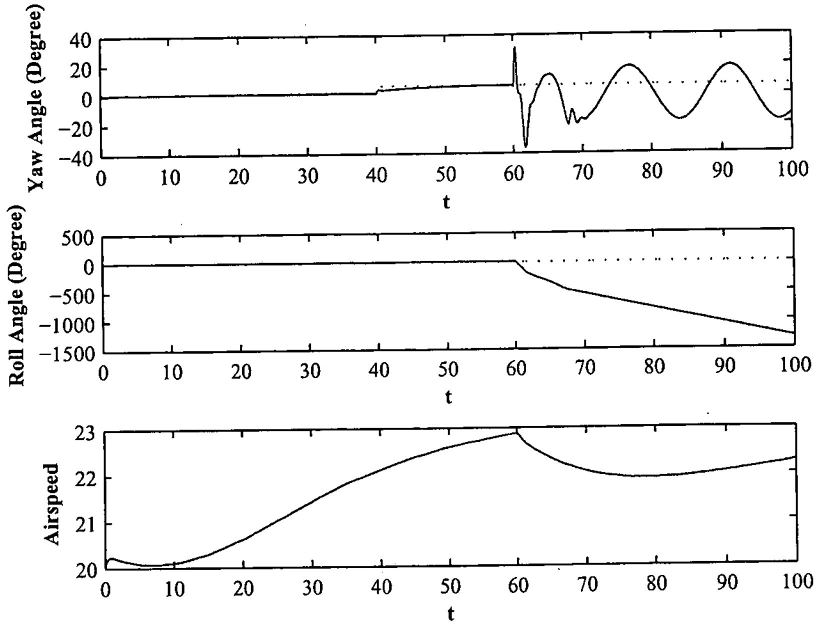


Figure 4.139: Aerosonde states corresponding to a hard over fault in the aileron

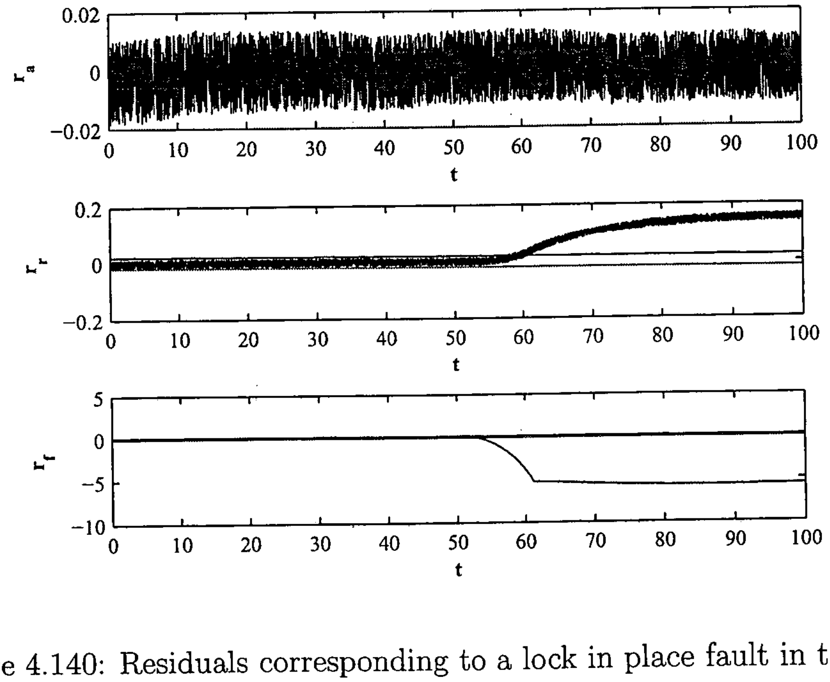


Figure 4.140: Residuals corresponding to a lock in place fault in the aileron

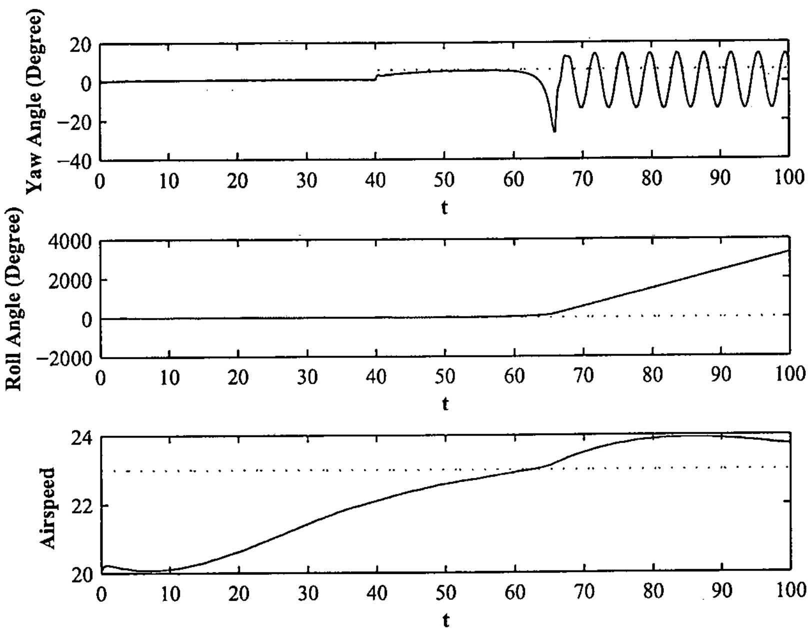


Figure 4.141: Aerosonde states corresponding to a lock in place fault in the aileron

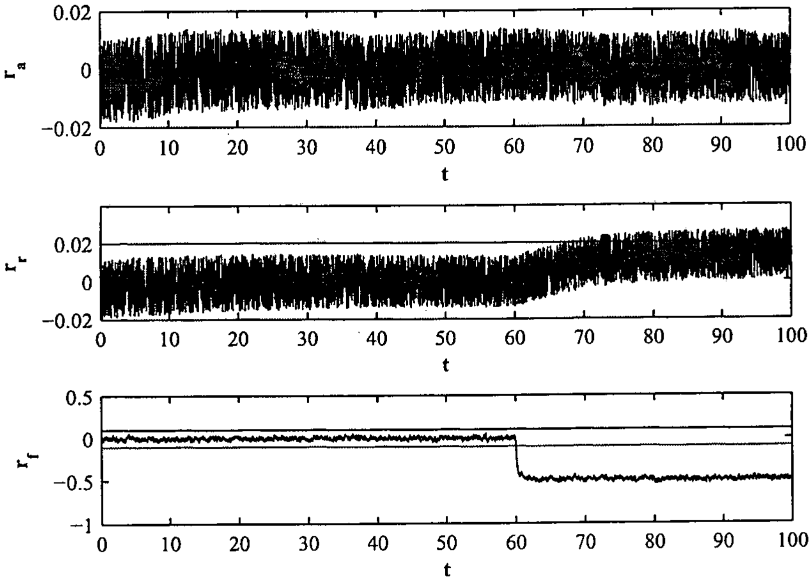


Figure 4.142: Residuals corresponding to a 20% loss of effectiveness fault in the aileron

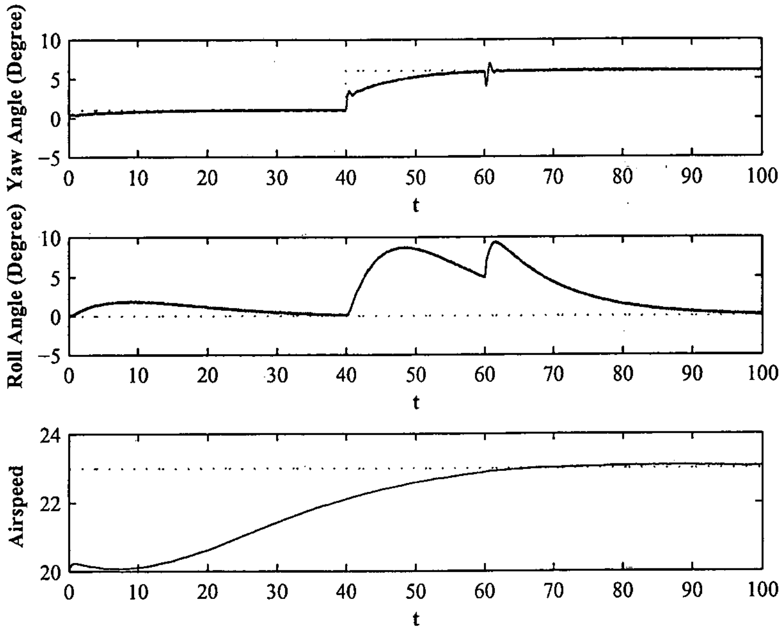


Figure 4.143: Aerosonde states corresponding to a 20% loss of effectiveness fault in the aileron

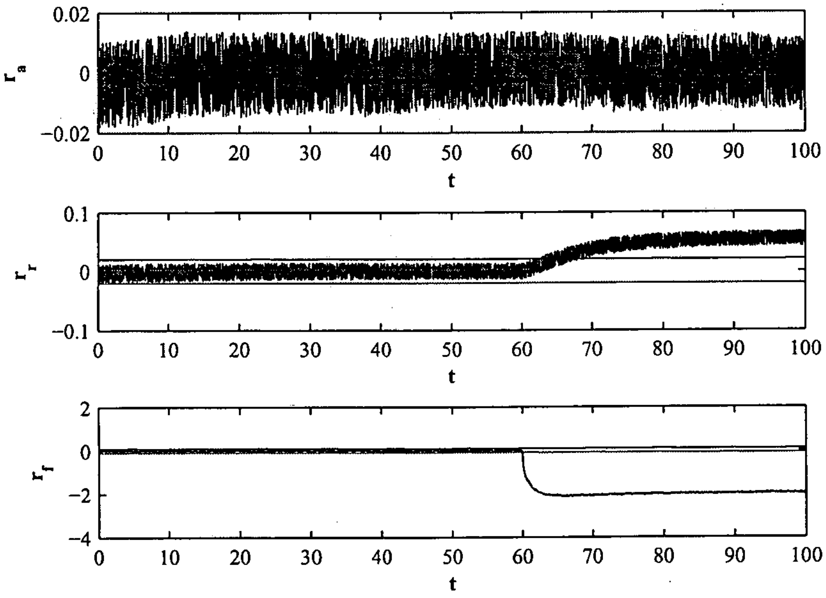


Figure 4.144: Residuals corresponding to a 50% loss of effectiveness fault in the aileron

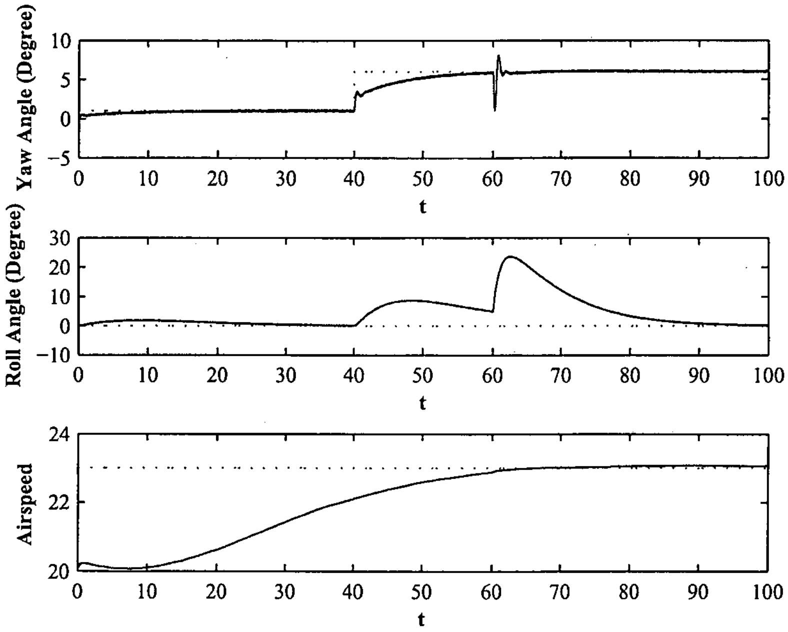


Figure 4.145: Aerosonde states corresponding to a 50% loss of effectiveness fault in the aileron

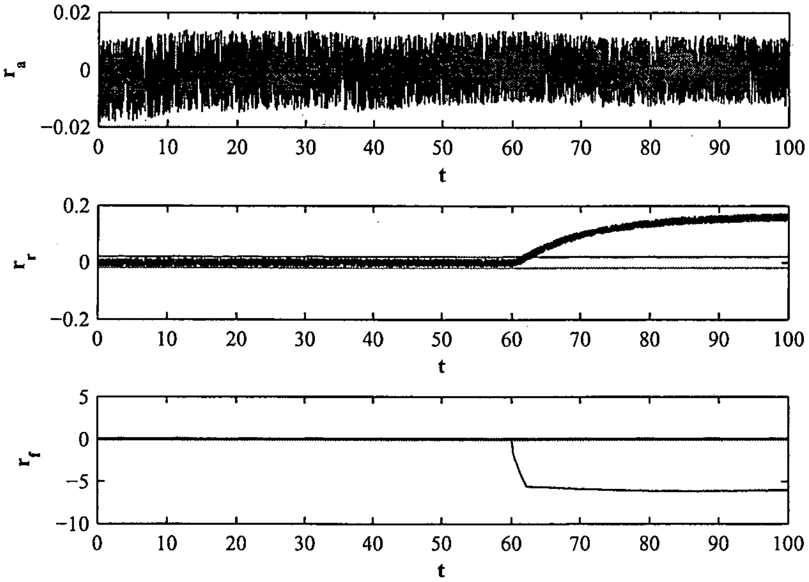


Figure 4.146: Residuals corresponding to a 80% loss of effectiveness fault in the aileron

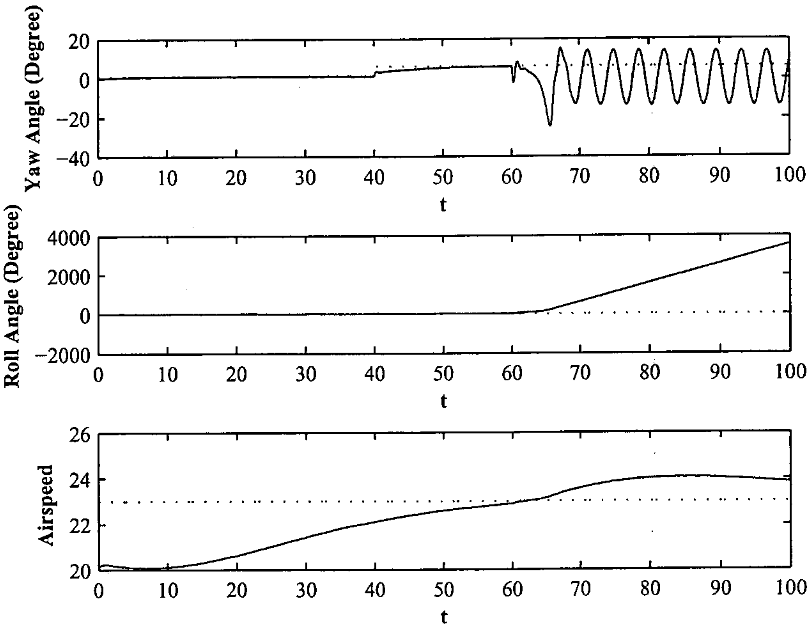


Figure 4.147: Aerosonde states corresponding to a 80% loss of effectiveness fault in the aileron

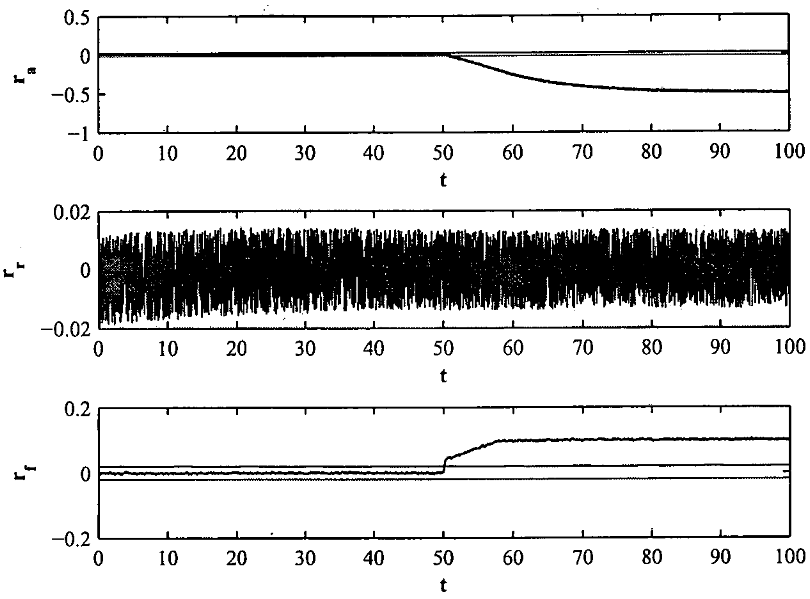


Figure 4.148: Residuals corresponding to a float fault in the rudder

Figure 4.148 shows the residuals corresponding to a float fault in the rudder. As shown in this figure the residuals r_a and r_F exceed their thresholds while residual r_r remains below the threshold. According to table 6.2 one can detect the occurrence of the fault in the rudder. Figure 4.149 demonstrates the states of aerosonde in this faulty scenario which implies that the aerosonde cannot track the yaw reference trajectory but it will remain stable. Figures 4.150 and 4.152 show the residuals corresponding to a hard over and lock in place fault in the rudder respectively. According to these figures and table 6.2 , one can easily detect and isolate these faults in the rudder. As shown in figures 4.151 and 4.153 , in both scenarios the aerosonde remains stable but cannot track the yaw reference trajectory. Figure 4.154 depicts the residuals corresponding to a 20% loss of effectiveness fault in the rudder. According to this figure one can only detect the fault but cannot isolate between rudder and engine. Moreover, as shown in figure 4.155, the controller compensates this fault and there exist just some transient effect on the aerosonde states. Figures 4.156 and 4.158 show the residuals corresponding to the 50% and 80% loss of effectiveness in the rudder respectively. In these two scenarios also, one can detect and isolate the fault in the rudder. Figures 4.157 and 4.159 depict the states of the aerosonde for 50% and 80% loss of effectiveness in the rudder respectively. According to these figures, the controller can recover from the 50% loss of effectiveness fault, while the 80% loss of effectiveness causes the steady state error in yaw angle.

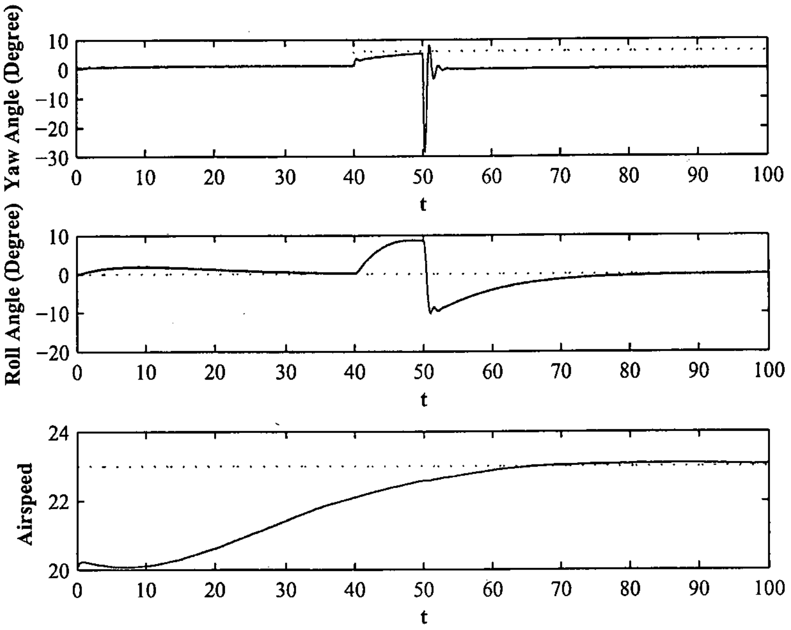


Figure 4.149: Aerosonde states corresponding to a float fault in the rudder

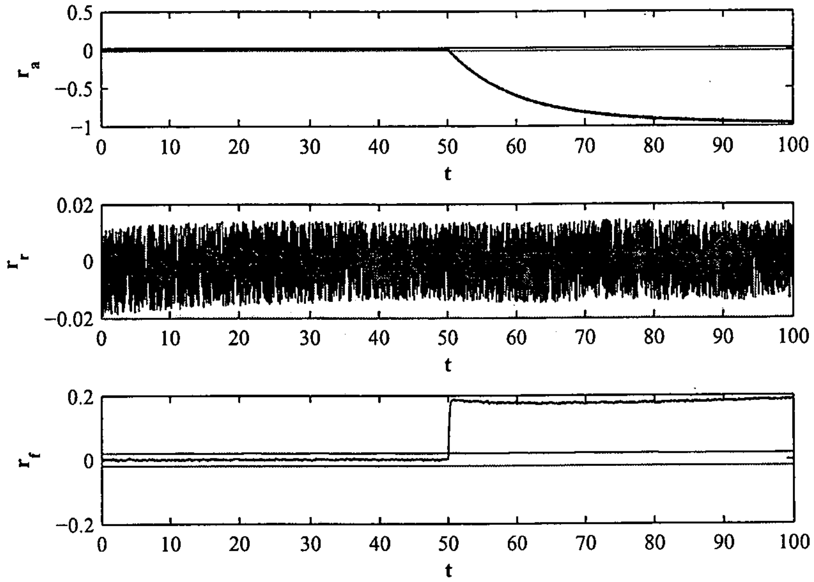


Figure 4.150: Residuals corresponding to a Hard over fault in the rudder

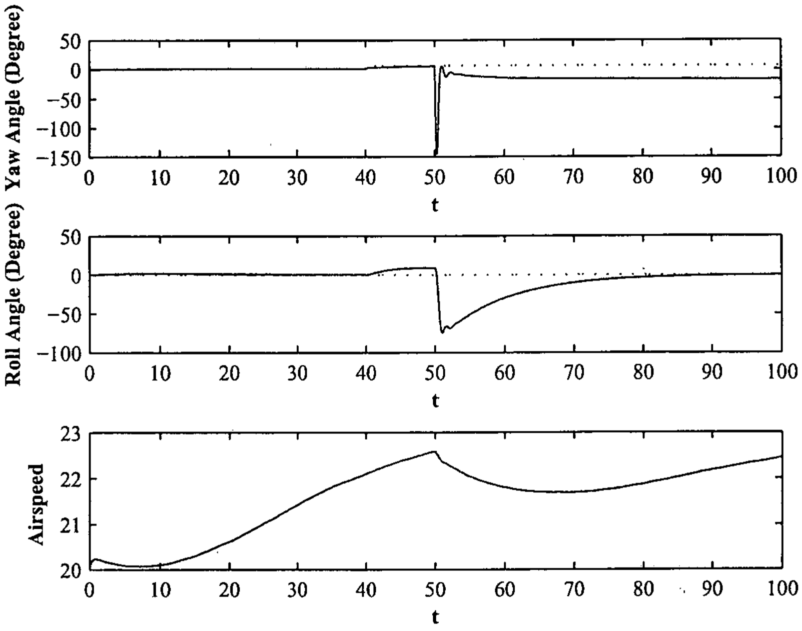


Figure 4.151: Aerosonde states corresponding to a hard over fault in the rudder

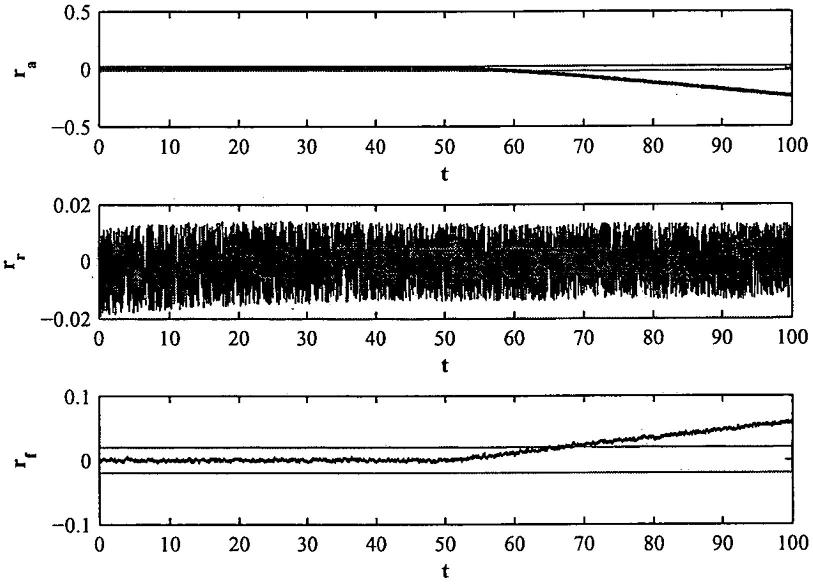


Figure 4.152: Residuals corresponding to a lock in place fault in the rudder

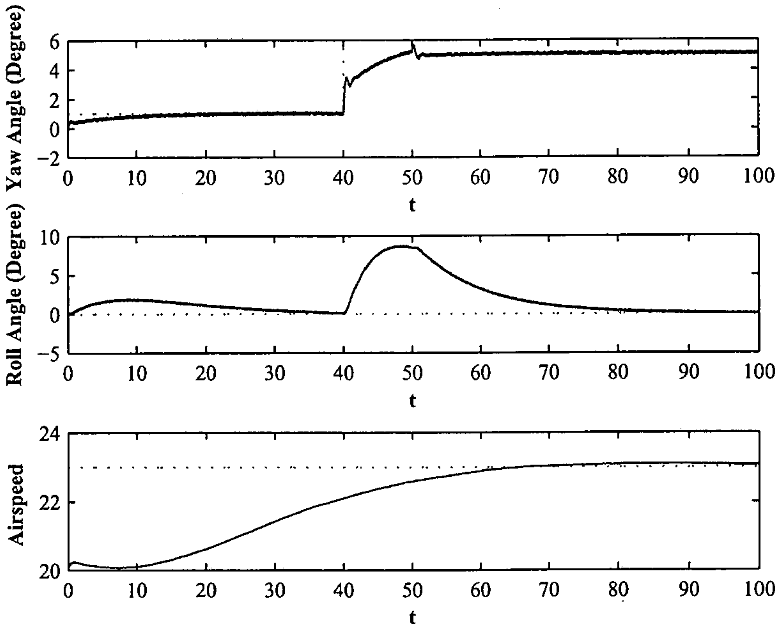


Figure 4.153: Aerosonde states corresponding to a lock in place fault in the rudder

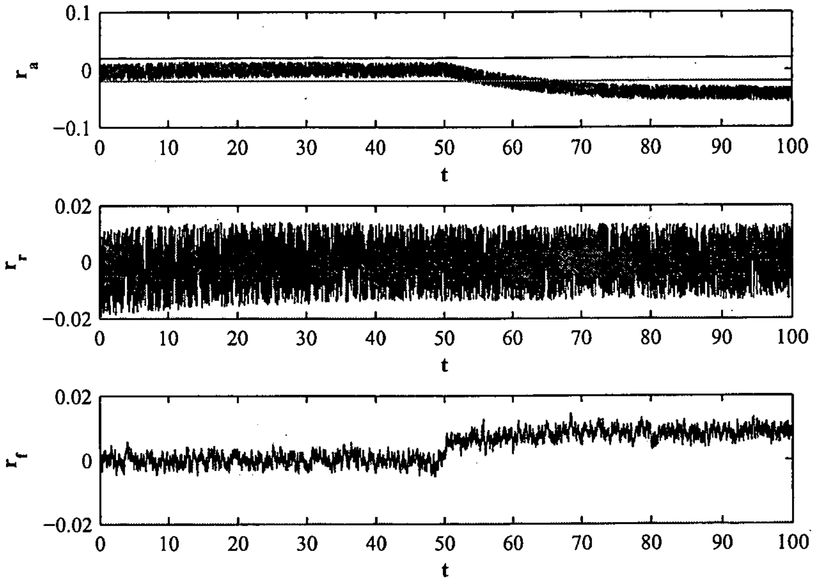


Figure 4.154: Residuals corresponding to a 20% loss of effectiveness fault in the rudder

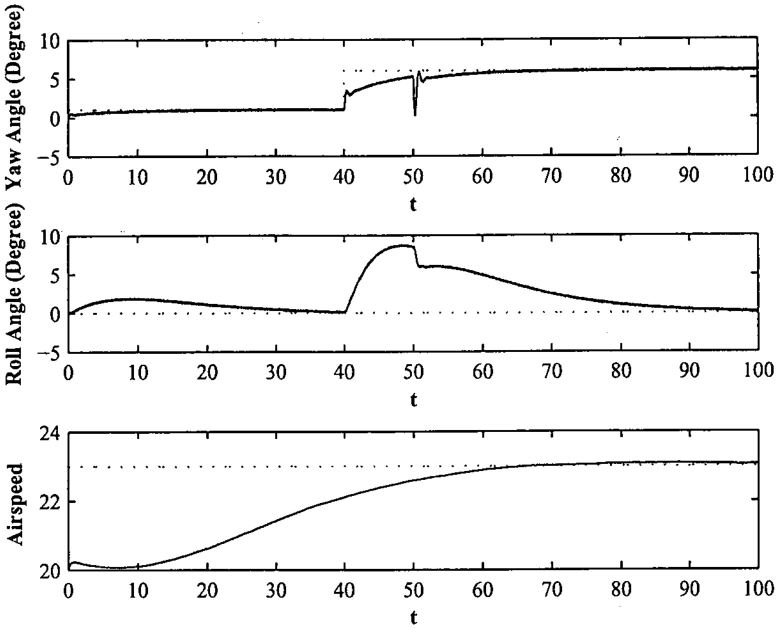


Figure 4.155: Aerosonde states corresponding to a 20% loss of effectiveness fault in the rudder

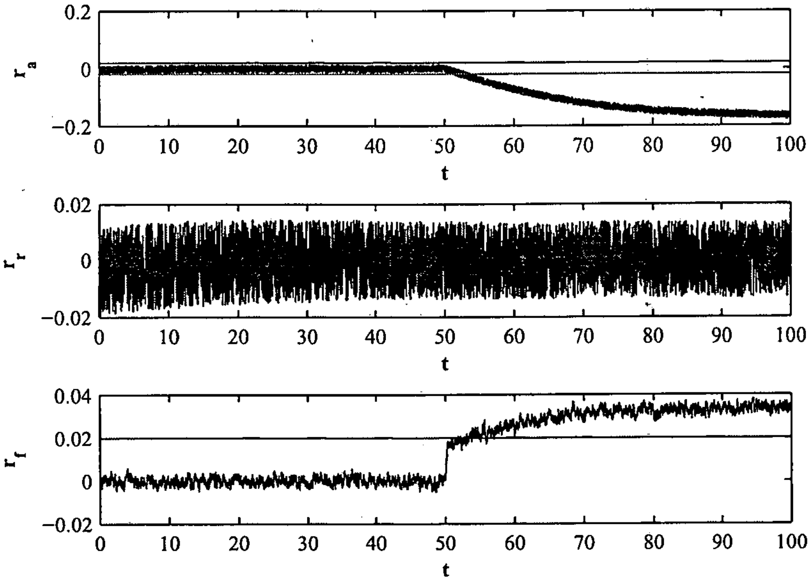


Figure 4.156: Residuals corresponding to a 50% loss of effectiveness fault in the rudder

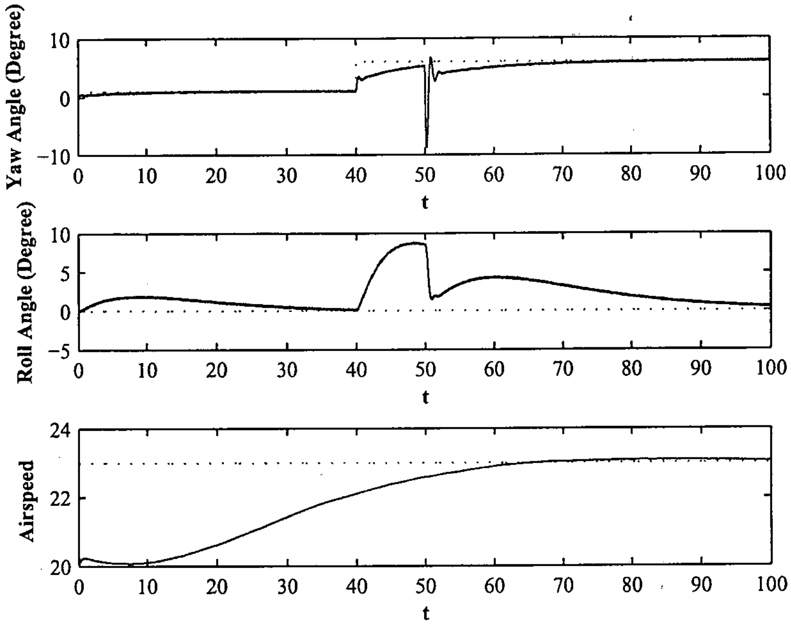


Figure 4.157: Aerosonde states corresponding to a 50% loss of effectiveness fault in the rudder

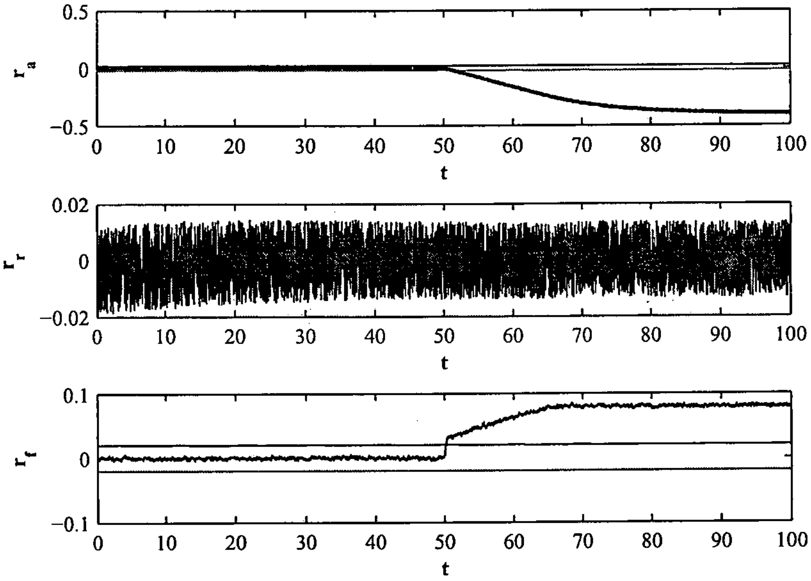


Figure 4.158: Residuals corresponding to a 80% loss of effectiveness fault in the rudder

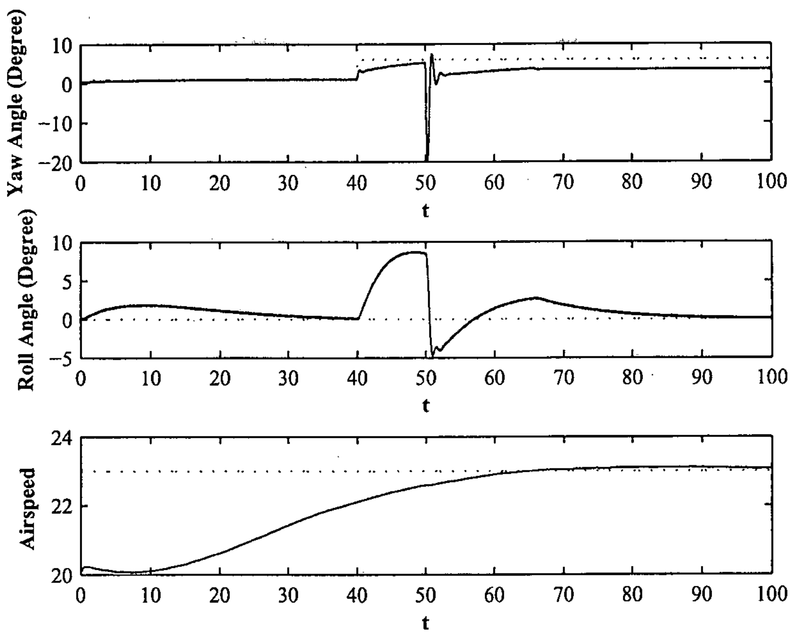


Figure 4.159: Aerosonde states corresponding to a 80% loss of effectiveness fault in the rudder

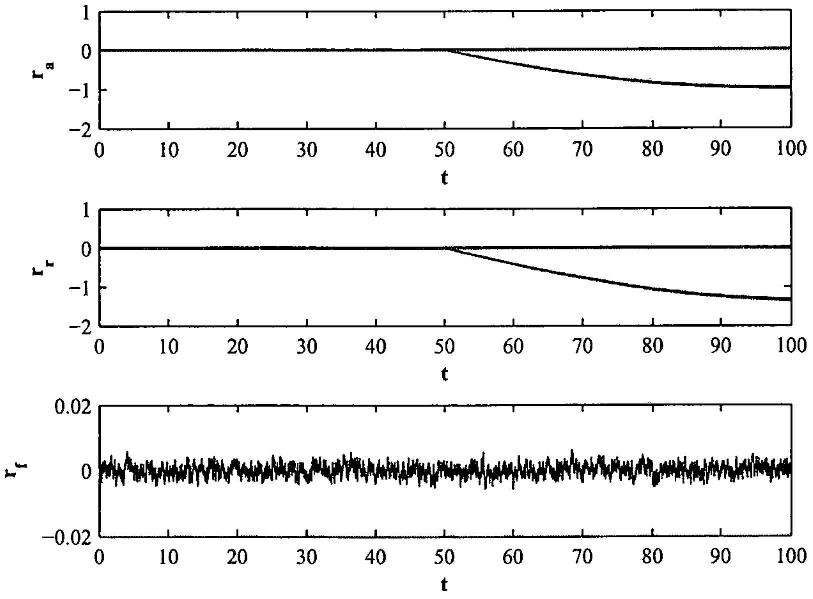


Figure 4.160: Residuals corresponding to a float fault in the engine

Figure 4.160 shows the residuals corresponding to a float fault in the engine. As shown in these figure the residuals r_a and r_r exceed their thresholds while residual r_F remains below the threshold. According to table 6.2 one can detect the occurrence of the fault in the engine. Figure 4.161 demonstrates the states of aerosonde in this faulty scenario which implies that the aerosonde cannot track the yaw and speed reference trajectories but it will remain stable. Figure 4.162 shows the residuals corresponding to a hard over and lock in place fault in the engine. According to this figure and table 6.2 , one can easily detect and isolate this faults in the engine. As shown in figure 4.163 the aerosonde remains stable but cannot track the speed reference trajectory. Figure 4.164 depicts the residuals corresponding to a 20% loss of effectiveness fault in the engine. According to this figure one can detect and isolate the fault in the engine. Moreover, as shown in figure 4.165, the controller can recover from this fault. Figures 6.6 and 4.168 show the residuals corresponding to the 50% and 80% loss of effectiveness in the engine respectively. In these two scenarios also, one can detect and isolate the fault in the engine. Figures 6.5 and 4.169 depict the states of the aerosonde for 50% and 80% loss of effectiveness in the engine respectively. According to these figures, in both cases, the aerosonde remains stable but there exist the steady state error in the speed.

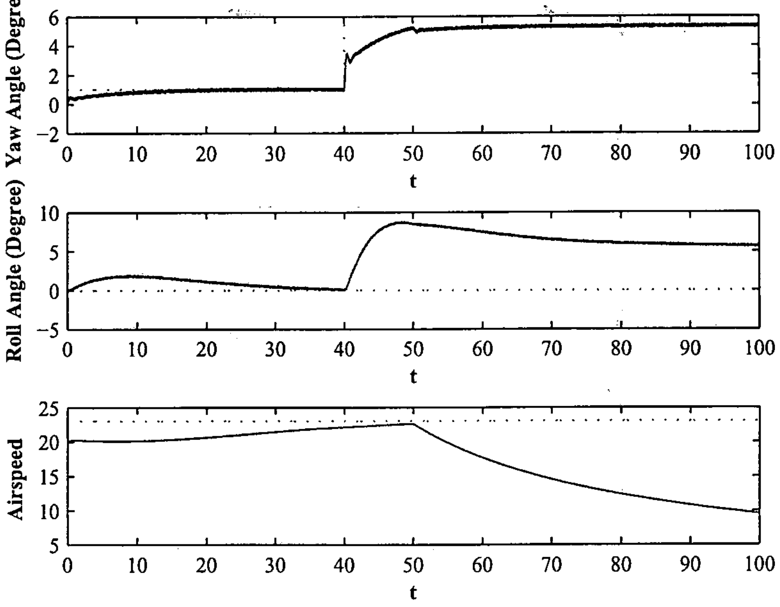


Figure 4.161: Aerosonde states corresponding to a float fault in the engine

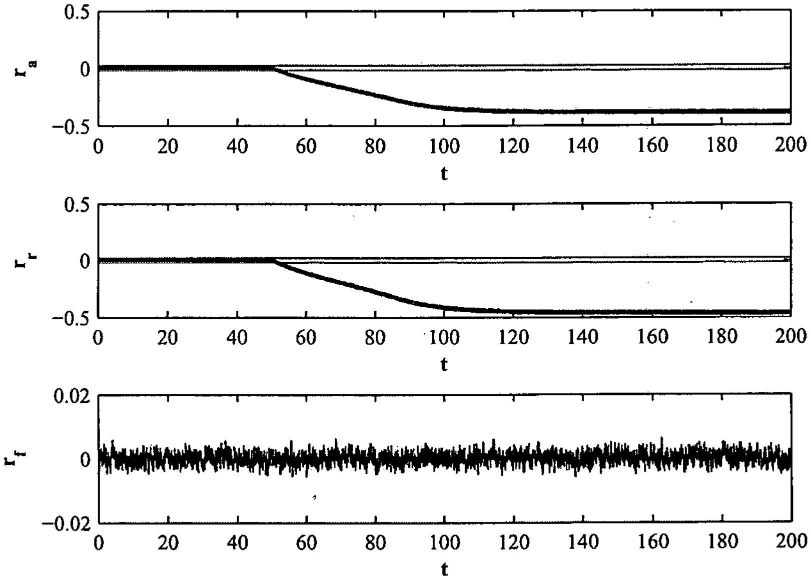


Figure 4.162: Residuals corresponding to a hard over fault in the engine

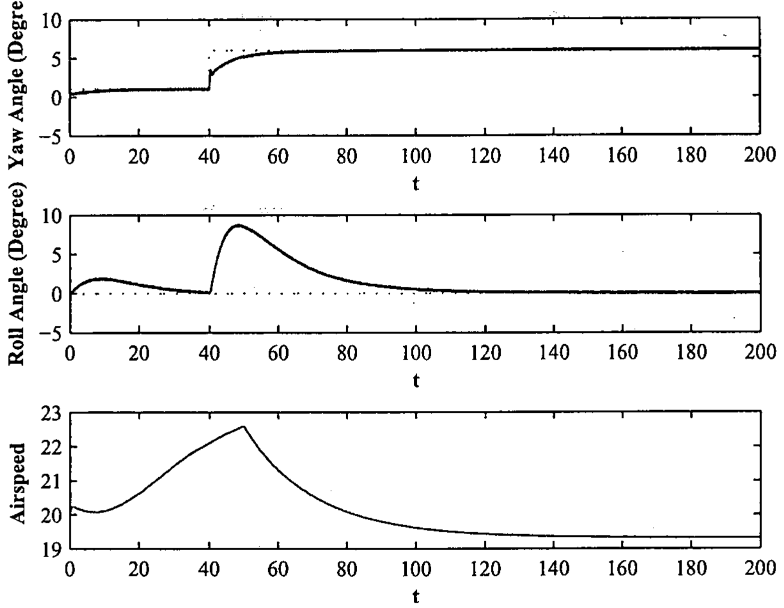


Figure 4.163: Aerosonde states corresponding to a hard over fault in the engine

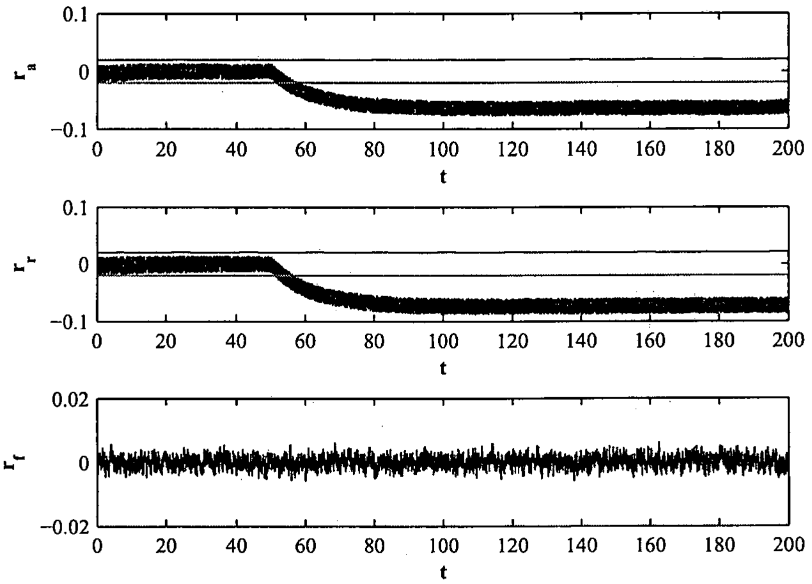


Figure 4.164: Residuals corresponding to a 20% loss of effectiveness fault in the engine

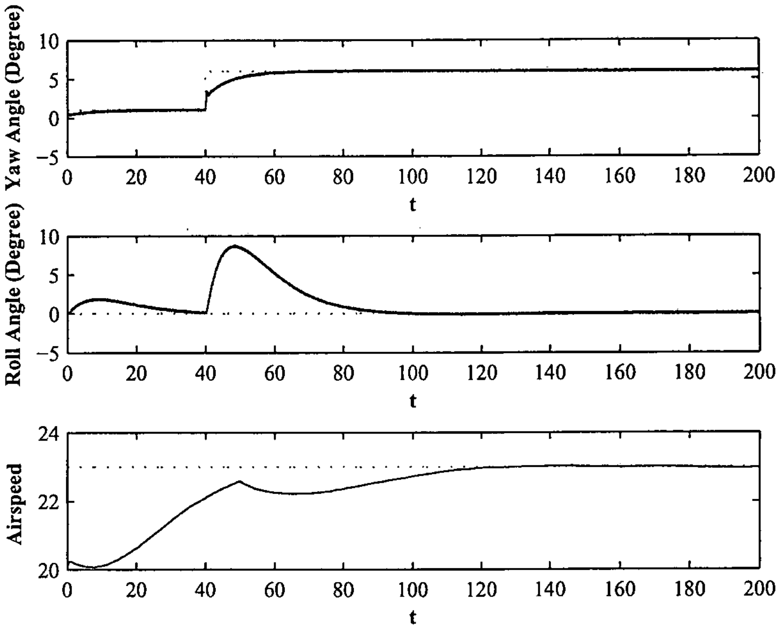


Figure 4.165: Aerosonde states corresponding to a 20% loss of effectiveness fault in the engine

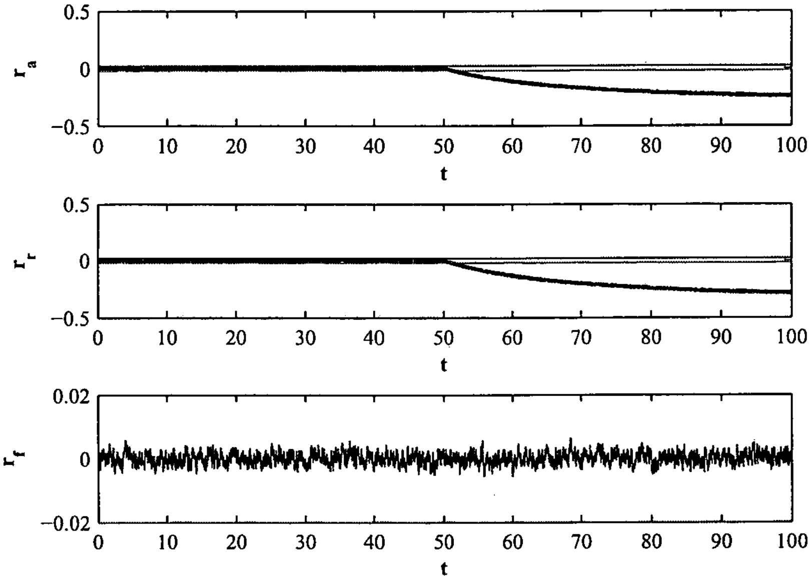


Figure 4.166: Residuals corresponding to a 50% loss of effectiveness fault in the engine

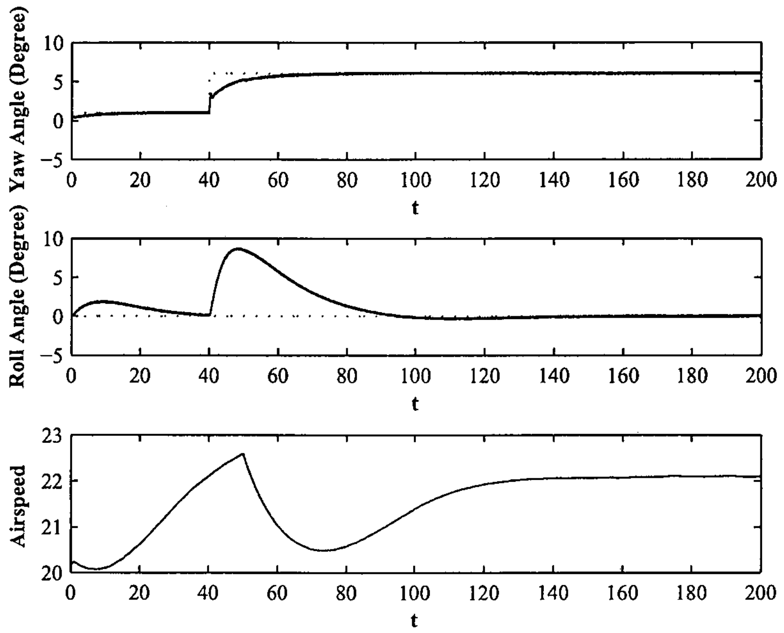


Figure 4.167: Aerosonde states corresponding to a 50% loss of effectiveness fault in the engine

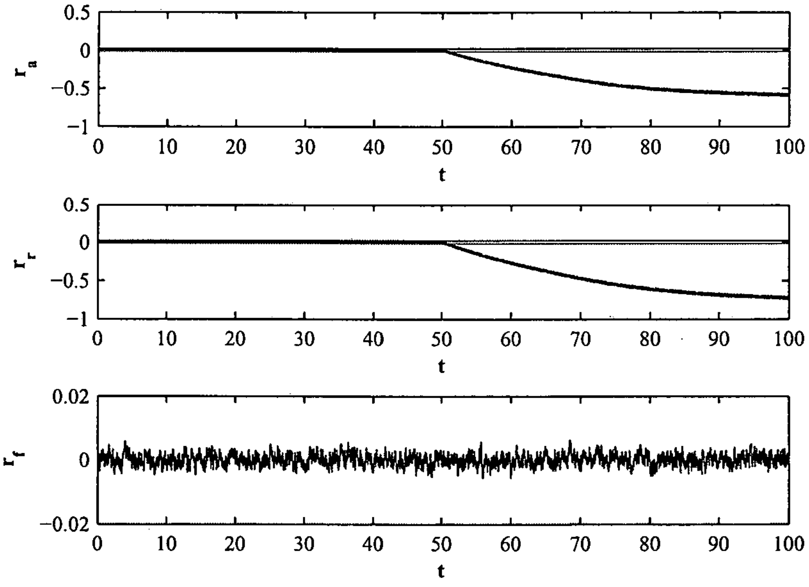


Figure 4.168: Residuals corresponding to a 80% loss of effectiveness fault in the engine

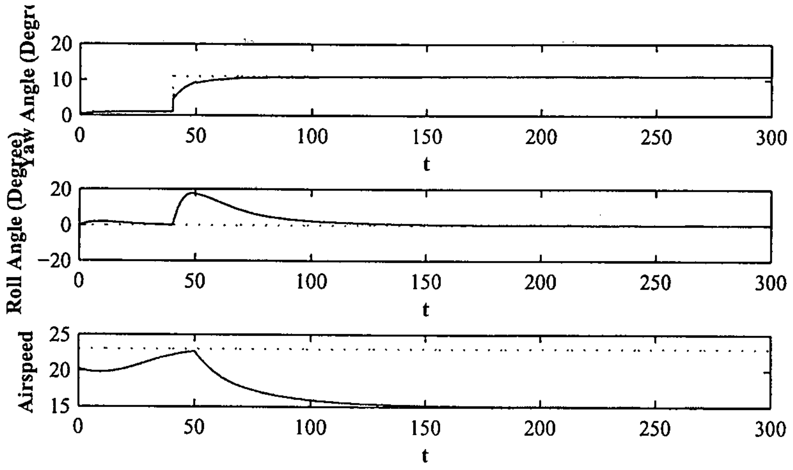


Figure 4.169: Aerosonde states corresponding to a 80% loss of effectiveness fault in the engine

Chapter 5

Neural Network Based FDIR

5.1 Introduction to Robust Fault Detection and Isolation

Model-based fault diagnosis (FD) approaches rely on the key assumption that a perfectly accurate and complete mathematical model of the system under supervision is available. However, such assumption is usually not valid in practice since it is difficult to obtain the necessary modeling accuracy required for construction of reliable analytical redundancy-based FD architectures. Unavoidable uncertainties that arise due to modeling errors, parameter variations, time variations, unknown external disturbances, and measurement noises deteriorate the performance of the FD schemes by causing false alarms. This performance deterioration can happen to an extent that makes the model-based FD scheme totally useless. This necessitates the development of FD algorithms, which have the ability to reliably detect as well as isolate faults and failures in the presence of modeling uncertainties. Such algorithms are referred to as *robust fault diagnosis* schemes.

To overcome the difficulties introduced by modeling uncertainties, a model-based FDI has to be made robust, i.e. insensitive to modeling uncertainty [55]. However, sometimes, merely reducing the sensitivity to modeling uncertainties does not solve the problem because such a sensitivity reduction may be undesirably accompanied by a reduction of the sensitivity to faults. Thus, a more meaningful formulation of the robust FDI problem is to increase the robustness to modeling uncertainty without losing sensitivity to faults. In conclusion, an FDI scheme designed to provide satisfactory sensitivity to faults, associated with the necessary robustness with respect to modeling uncertainty, is called *a robust FDI scheme.*

The importance of robustness in model-based FDI has been widely recognized by both academia and industry. More specifically, robust FDI for linear systems has been extensively investigated by many researchers during the last two decades. As a result, a number of methods have been proposed to tackle the linear robust FDI problem [55] such as the unknown input observer (UIO) method [56], eigen-structure assignment [57], and optimally robust parity relation methods [58]. However, the problem of robust FDI for nonlinear systems has not been investigated as extensively as its linear counterpart. More specifically, when the isolation of the faults in the nonlinear system is required rather than just detecting them, very few works have been reported in the literature.

Traditionally, the robust FDI problem for nonlinear dynamic systems has been approached in two steps. Firstly, the model is linearized at an operating point, and then robust linear FDI techniques are applied to generate residuals that are insensitive to modeling uncertainties but responsive to faults. This method only works well when the linearization does not cause a large mismatch between linear and nonlinear models and the system operates close to the operating point specified. As another alternative to robust nonlinear FDI, one might think of just simply increasing the threshold levels of the residuals generated by the nonlinear FDI scheme and thus reducing the number of false alarms. However, this increase in the threshold levels will at the same time decrease the fault sensitivity of the FDI scheme. This imposes a tradeoff between reducing the number of false alarms and missing to detect the presence of an actually occurred fault. A reliable solution to such a trade-off problem is not trivial in practice specially due to the nonlinear behavior of system dynamics and the presence of different sources of unknown uncertainties. Therefore, there is a high demand for development of techniques that make the nonlinear FDI problem robust to modeling uncertainties, that is, remarkably reducing the number of false alarms when the nonlinear system is under healthy mode of operation, whilst reliably detecting the presence of faults or failures.

Robustness in the analytical redundancy-based framework to FDI is usually achieved by either making the *residual generation* process or the *residual evaluation* process insensitive to modeling uncertainties. In this work we will take the former viewpoint and propose, design and develop a residual *generation* scheme that is robust to modeling uncertainties. We will use neural networks for this purpose due to their distinguished ability to implement and approximate, to an arbitrary level of accuracy, any continuous nonlinear static or dynamic function, given suitable network parameters (or weights), architecture and learning algorithm [59]. Neural networks have been widely used in many engineering domains such as nonlinear control [60] [61], nonlinear system identification [60], and nonlinear state estimation [62] [63] and the fault diagnosis application field is by no means an exception. There have been a large number of publications on neural networks-based FDI, e.g. [64] [65] [66] [67]. However, the major shortcomings of these works are the negligence of the possible presence of mathematical models of the system and consequently the lack of formal and analytical results on the performance of the FDI subsystem. Furthermore, they are mainly focused on the fault detection problem rendering the fault isolation issue unsolved.

The salient feature of our robust FDI scheme is that the neural network capabilities in dealing with unknown nonlinearities are exploited as a robustifying tool on top of the nonlinear geometric FDI approach, which was presented in the previous sections. This has two main advantages:

(i) The very powerful capabilities of the nonlinear geometric approach in fault isolation, as shown previously, will be exploited. Thus, we don't need to design an extra FDI algorithm.

(ii) Existence of analytical proofs for model-based approaches like nonlinear geometric FDI make the whole FDI scheme analytically tractable and thus more reliable.

In conclusion, the proposed robust nonlinear FDI scheme will be able to simultaneously benefit, within a *hybrid* framework, from the existence of mathematical models and highly reliable performance of model-based geometrical FDI method, and the superior capabilities of neural networks in estimating unknown nonlinear dynamic functions. In sequel, the details of the proposed robustifying scheme are provided.

5.2 Hybrid Framework to Robust FDI

The model-based nonlinear geometric FDI scheme, presented in chapter 4, is depicted in Figure 5.1 where the appropriate state and output transformations have to be determined leading to a quotient subsystem unaffected by all faults but one. Hence, the state and output transformations and residual generators (nonlinear observers, etc.) have to be designed for each possible fault. In the ideal scenario, for a particular fault, the corresponding residual should remain close to "zero" when the corresponding fault is not present, and remain above "zero" when the fault occurs. Highly reliable performance of the geometric FDI in detecting and isolating different types of faults was verified using numerous simulation results. It was assumed that the mathematical model of the system used for the design of state transformation and observers completely matches the real system dynamics. However, any discrepancies between the mathematical model and the actual system, due to un-modeled dynamics and/or parameter uncertainties, will reflect themselves in the residuals generated by the geometric FDI scheme. Thus, even in the absence of faults in the system, the residuals will not remain close "zero"and will exceed their thresholds and consequently trigger a false alarm.

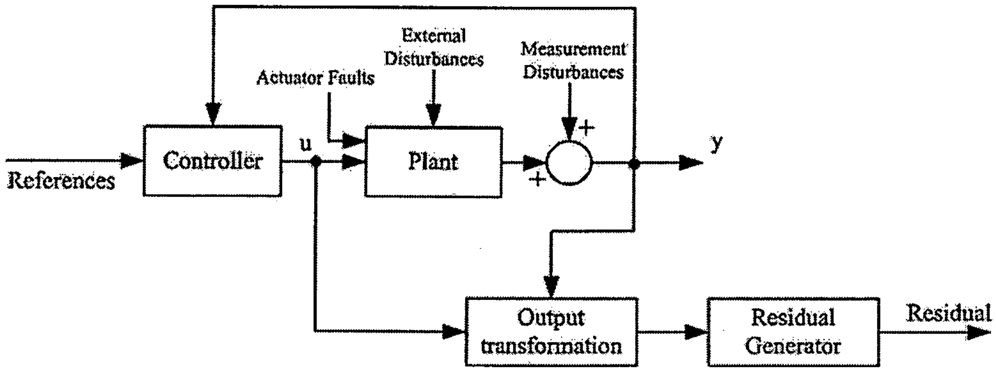


Figure 5.1: Model-based FDI diagram

Only a few number of publications in robust nonlinear fault diagnosis has been reported in the literature [68] [69] [70]. However, their emphasis is on fault detection and neural network-based fault severity estimation without addressing the problem of fault isolation. The novelty of our work is in the use of neural networks in conjunction with the mathematically proven nonlinear geometric FDI approach, which makes not only the detection but also the isolation of faults possible even in the presence of modeling uncertainties with minimum false alarms.

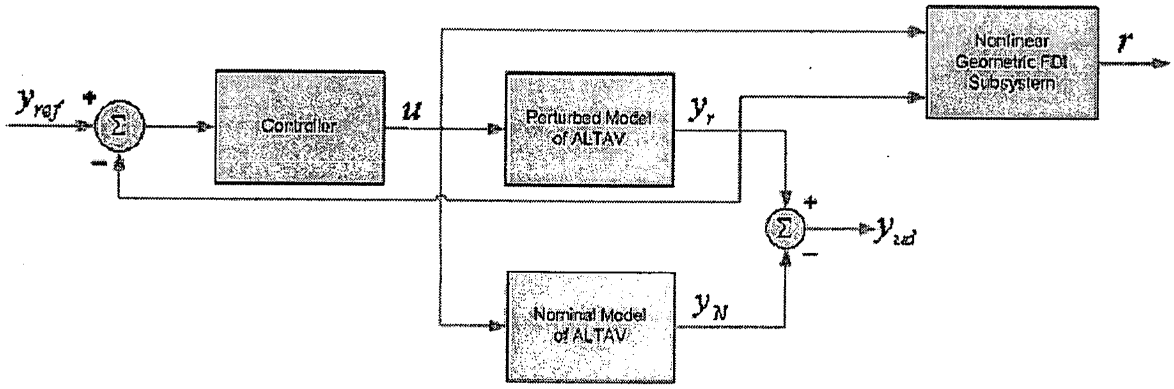


Figure 5.2: **Closed-loop simulation to evaluate modeling uncertainties effects on the FDI subsystem**

5.3 Effect of Uncertainties on the FDI Subsystem Residuals

Although the robust FDI scheme that we have developed in this project is general in nature, we have verified its applicability and performance only for ALTAV platform. Thus, consider the mathematical model of the ALTAV dynamics given in equation 4.35. To account for modeling uncertainties, we will introduce deviations in the parameters of the mathematical model of the system from their nominal values. In other words, we assume that the physical parameters of the actual system do not perfectly match their corresponding nominal values derived at the time of modeling. This is a viable assumption since exact determination of some of these parameters such as the inertias or aerodynamic drag is not practically feasible. We may name the mathematical model of the system with perturbed parameters, the *perturbed model* of the system. We need the notion of the perturbed model due to inaccessibility of the actual ALTAV platforms for robust FDI test and verification purposes. Clearly, the perturbed model can be considered as a mathematical representation of the actual system in the absence of the platforms even in the presence of parameter uncertainties.

Then, we will simulate the closed-loop system with both the nominal model and the perturbed model operating in parallel, while the FDI subsystem is receiving both the perturbed model outputs and the common control signals as inputs. The block diagram representation of such a simulation scheme is depicted in Figure 5.2.

The reason for running the nominal model in parallel to the perturbed model is just to evaluate the effect of the introduced parameter uncertainties on the system outputs. More precisely, the difference between the outputs of the nominal model and the perturbed model represents the effect of uncertainties.

The parameters of ALTAV, refer to equation (4.35), that have been perturbed from their nominal values are listed in Table 5.1 with the magnitude of perturbations/uncertainties stated in percentage for training. In this report, two testing scenarios will be considered, in which the parameters are described by Table 5.2.

Figures 5.3 to 5.14 show the simulation results for all the twelve outputs of the perturbed and the nominal model of the ALTAV with rectangular reference trajectory for the ALTAV.

Table 5.1: Parameter Uncertainties of Training Scenario

Parameters	Nominal Values	Amount of Uncertainties (%)
C_γ	0.5	15
C_ϕ	0.5	15
J_γ	1.005	10
J_ϕ	1.005	10
L_B	0.16	15

Table 5.2: Parameter Uncertainties of Testing Scenarios

Parameters		C_γ	C_ϕ	J_γ	J_ϕ	L_B
Nominal values		0.5	0.5	1.005	1.005	0.16
Amount of Uncertainties (%)	Test 1	0	0	0	0	0
	Test 2	30	30	20	20	30

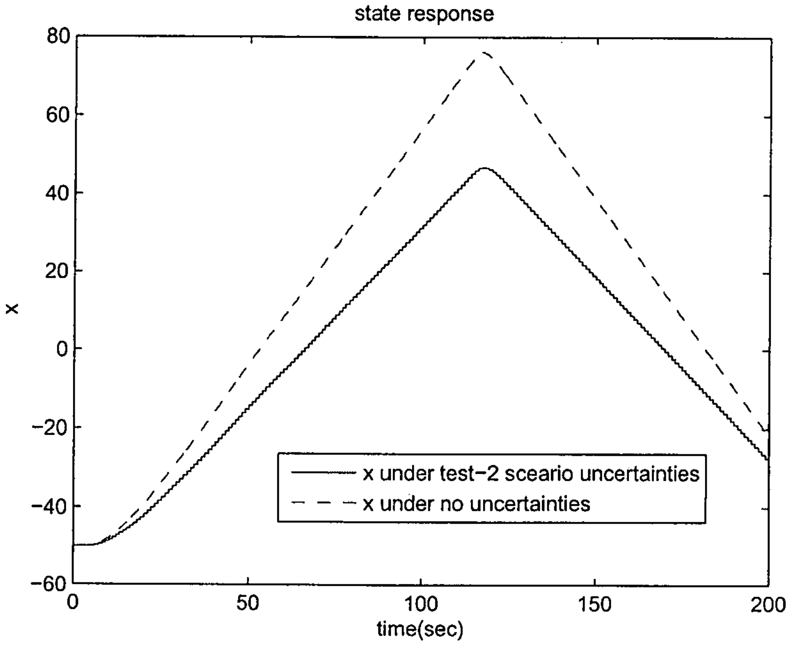


Figure 5.3: **state x with uncertainties**

Although the sensitivity of the outputs of the perturbed model of the ALTAV pretty much depends on the robustness of the ALTAV controller, it can be easily seen from the figures that the presence of uncertainties has generated a considerable impact on the some of output signals. This emphasizes the fact that the modeling uncertainties could cause significant discrepancies between the actual system behavior and the nominal model behavior that is utilized in the FDI design.

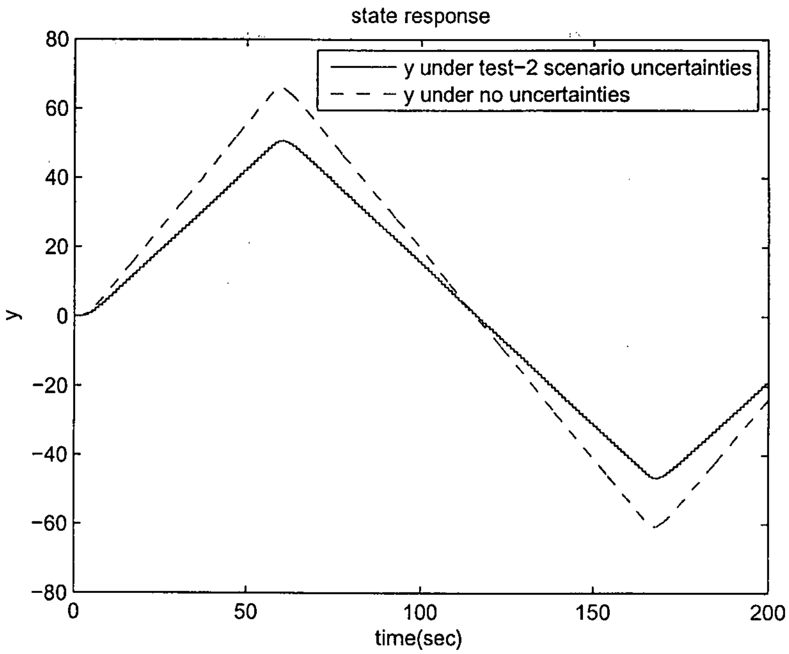


Figure 5.4: **state $\dot{x}$ with uncertainties**

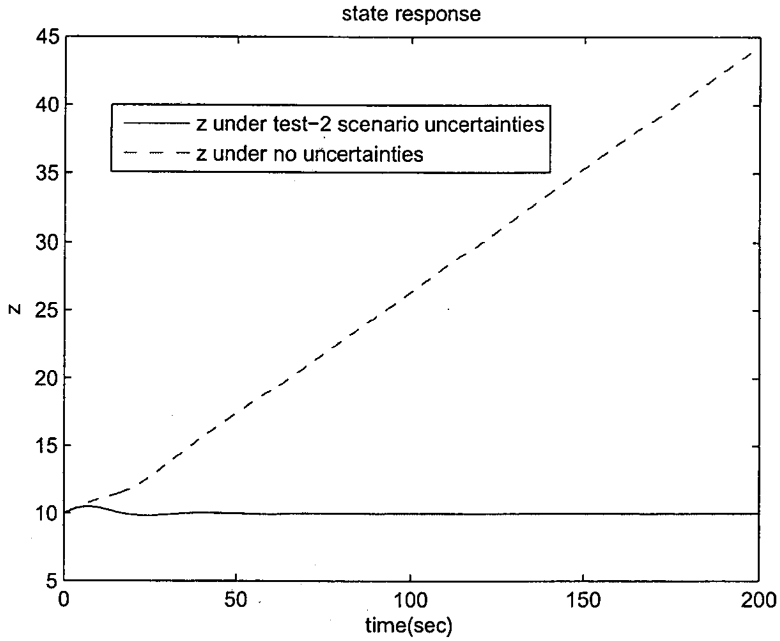


Figure 5.5: **state y with uncertainties**

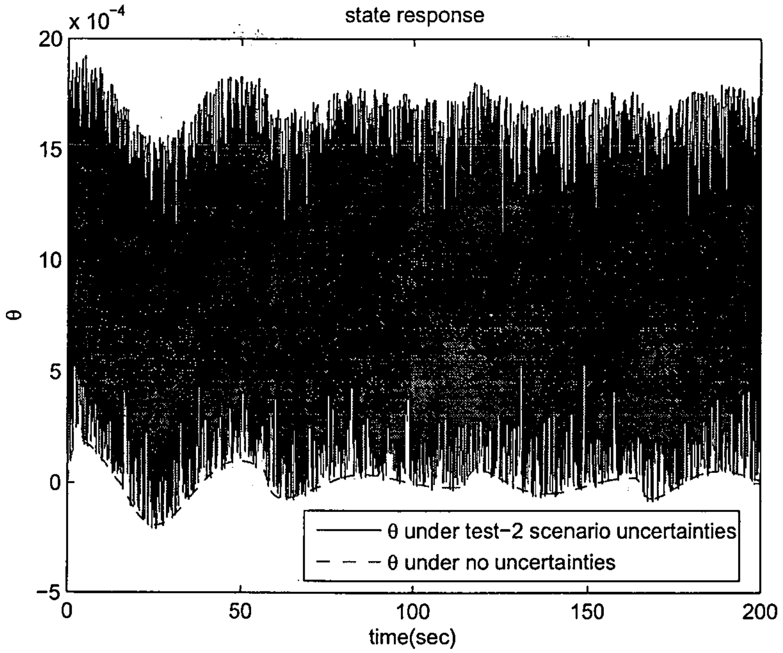


Figure 5.6: **state $\dot{y}$ with uncertainties**

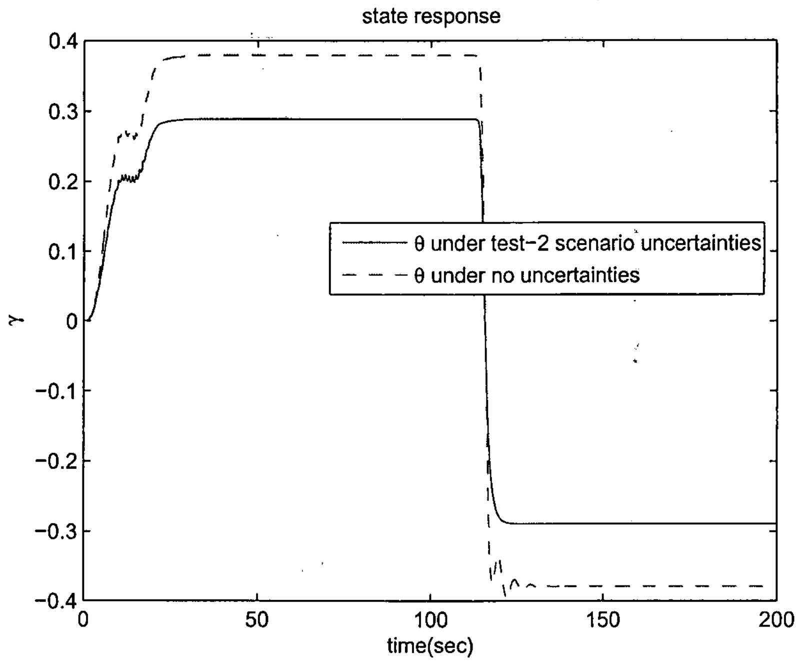


Figure 5.7: **state z with uncertainties**

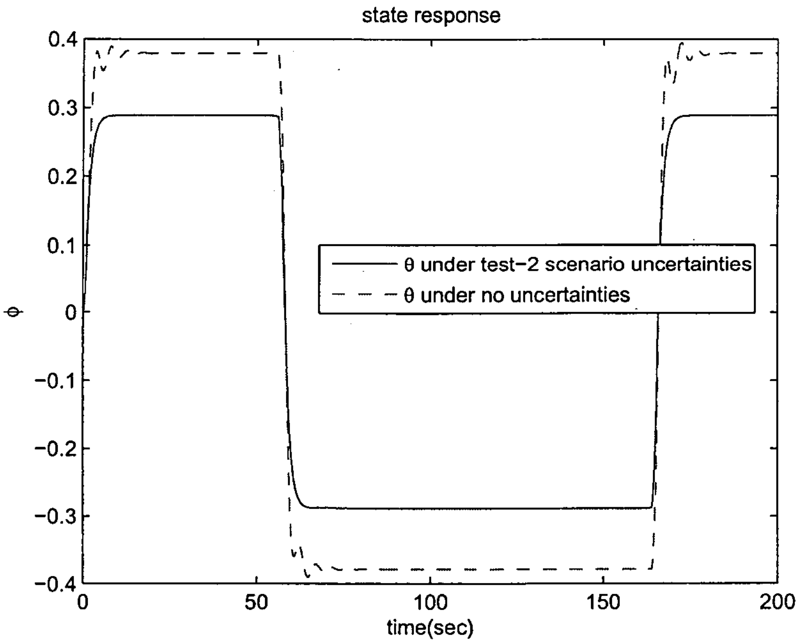


Figure 5.8: **state $\dot{z}$ with uncertainties**

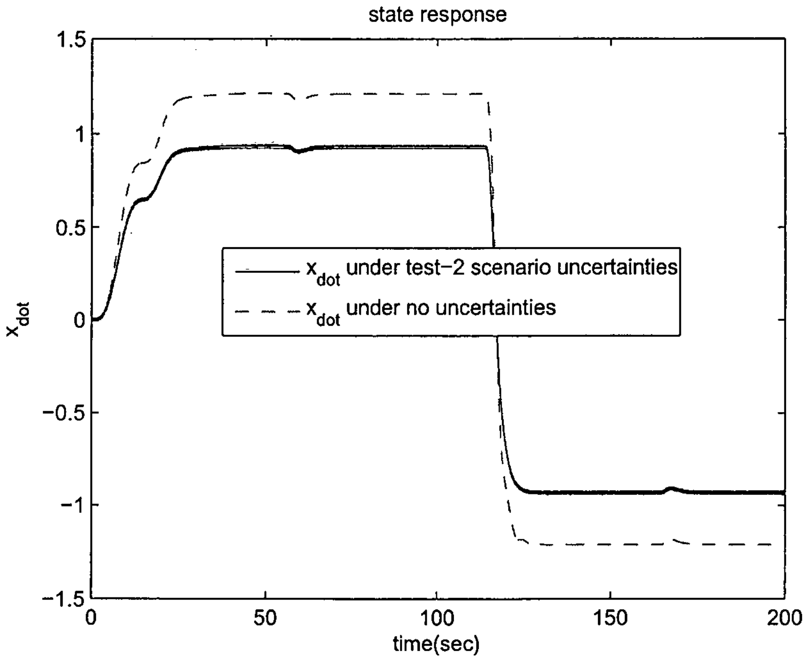


Figure 5.9: **state θ with uncertainties**

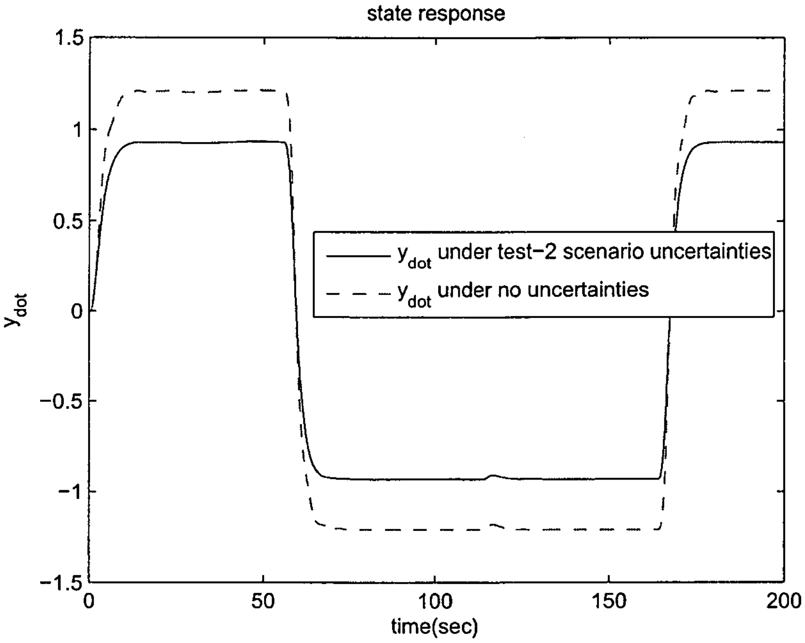


Figure 5.10: **state $\dot{\theta}$ with uncertainties**

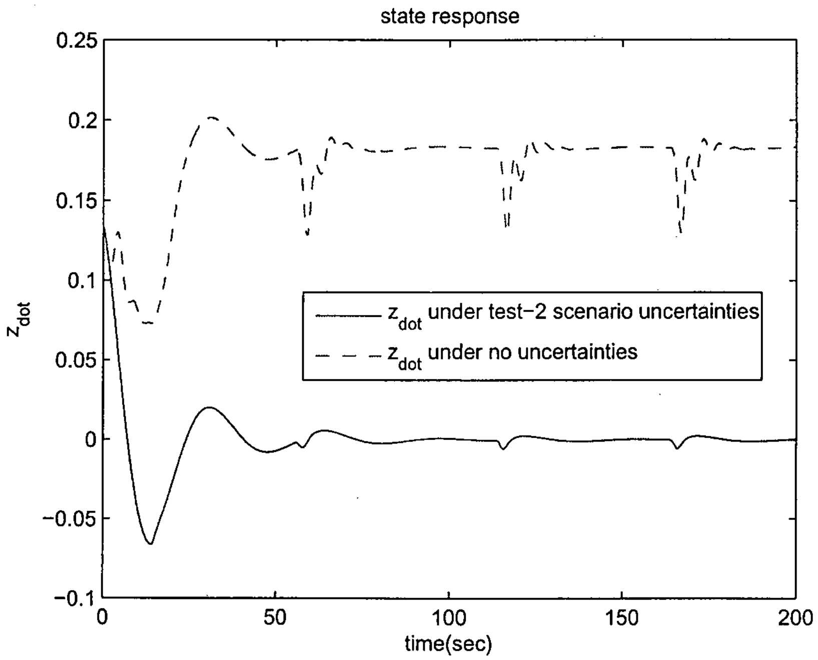


Figure 5.11: **state γ with uncertainties**

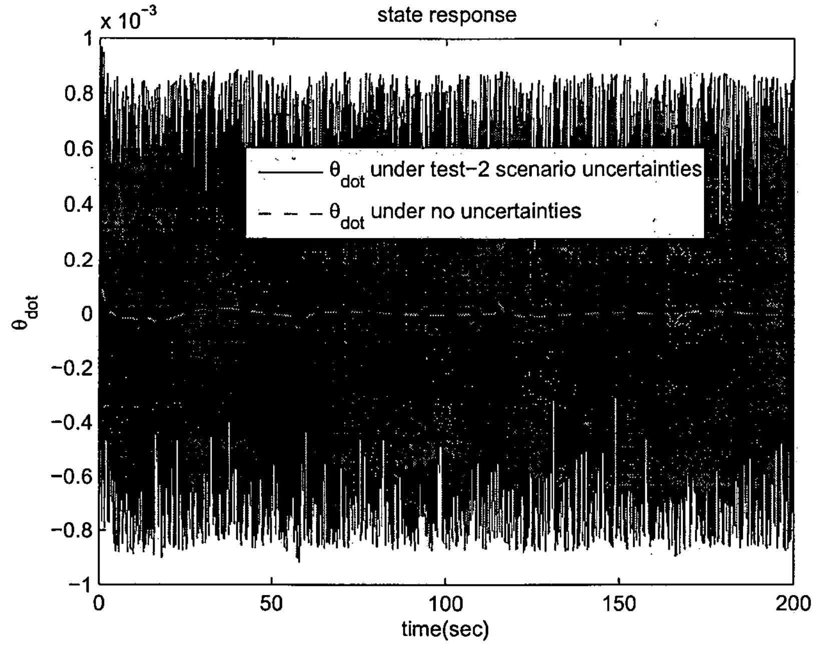


Figure 5.12: **state $\dot{\gamma}$ with uncertainties**

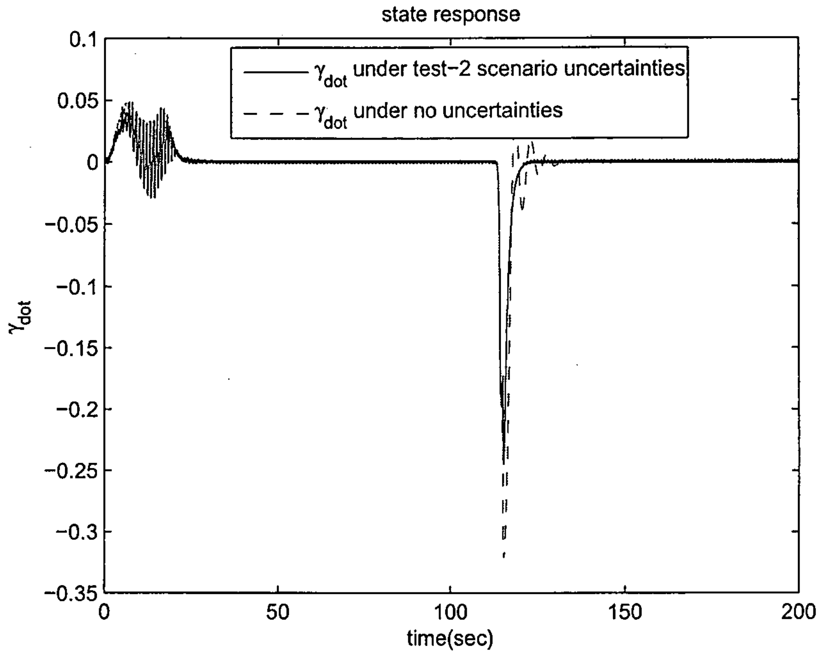


Figure 5.13: **state ϕ with uncertainties**

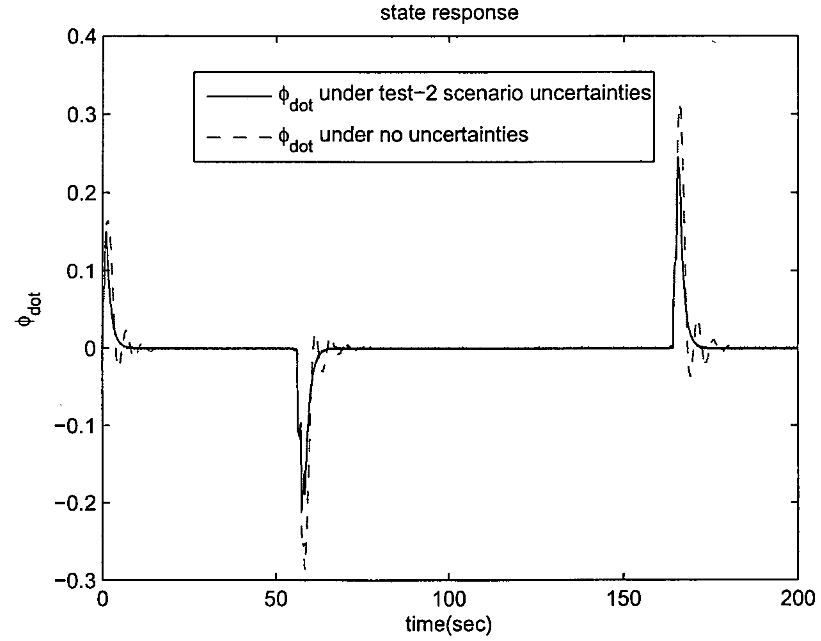


Figure 5.14: **state $\dot{\phi}$ with uncertainties**

The FDI subsystem residuals in the presence of uncertainties and in the absence of faults are also given in Figures 5.15. It is worth to emphasize that the perturbed model outputs, as opposed to the nominal model outputs, are fed to the FDI subsystem, as shown in Figure 5.2. This is quite reasonable since in real operational conditions the only available signals are the measured outputs of the actual ALTAV platform.

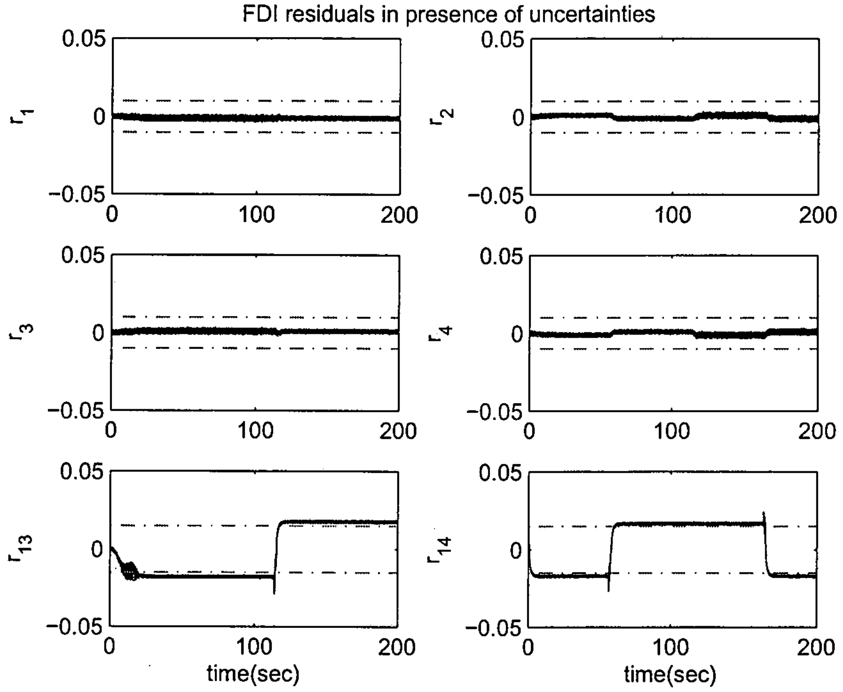


Figure 5.15: **All uncompensated residuals in the presence of modeling uncertainties (Test 2 scenario)**

As can be seen in Figures 5.16 and 5.17, in the presence of the introduced parameter uncertainties, at least one of the two residuals r_{13}, r_{24} are beyond their associated threshold bounds over a window of time, even though there is no fault in the system. This will cause the generation of *false alarms* by the FDI subsystem.

It should be noted that the return of the r_{13} residual to within the threshold bounds at around $t = 120(sec)$ is due to the nature of the reference trajectory of the ALTAV, which was set to rectangular as mentioned before. More precisely, after a complete turn from one side of the rectangle to the other perpendicular side, the value of the state γ and consequently the effect of introduced uncertainties on the states γ, $\dot{\gamma}$ become identically zero. Thus, the effect of uncertainties is not observable in the two states γ, $\dot{\gamma}$ during the travel of the ALTAV over two of the parallel sides of the rectangular reference trajectory. However, during the above-mentioned period of time, the FDI subsystem still observes the effect of modeling uncertainties through the r_{13} residual, thus again generating false alarms. Moreover, with a general reference trajectory, both residuals r_{13}, r_{24} will remain out of their bounds, both causing false alarms.

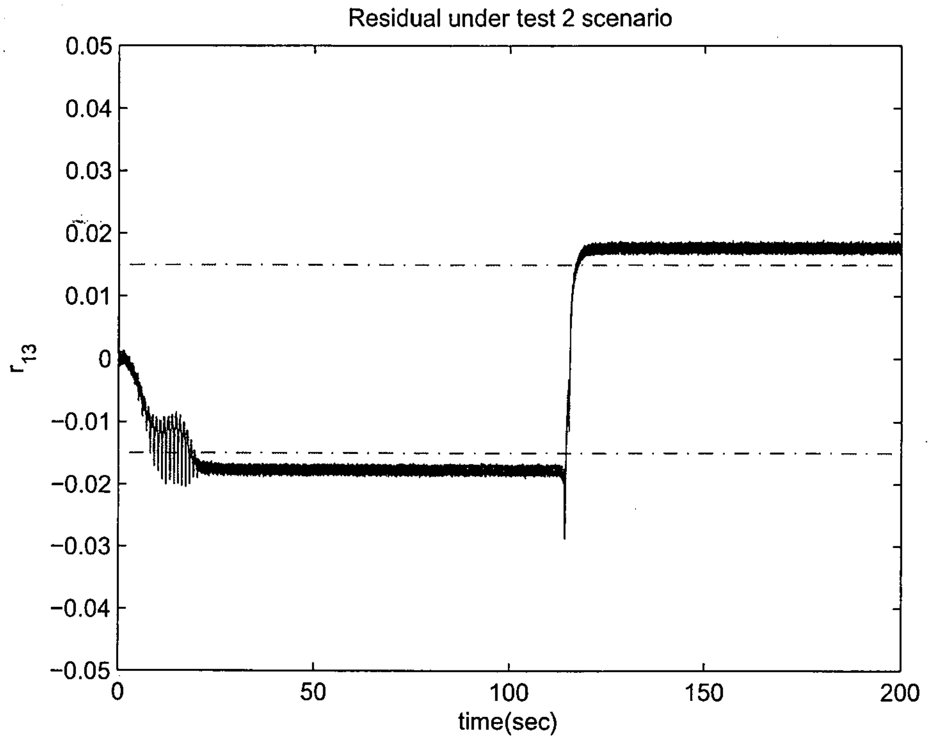


Figure 5.16: **Uncompensated residual r_{13} in the presence of modeling uncertainties (Test 2 scenario)**

5.3.1 Mathematical Representation of the Effect of Modeling Uncertainties

In this section, we will derive a state-space representation of the effect of uncertainties on the system outputs. Towards this end, let us consider the nominal model of a general nonlinear dynamical system with full state measurement and no uncertainties:

$$\begin{aligned} \dot{x}_N &= F_N(x_N, u) \\ y_N &= x_N \end{aligned} \tag{5.1}$$

where $x_N \in \Re^n$ is the state vector of the nominal system, $u \in \Re^m$ is the control vector and $F_N : \Re^n \times \Re^m \rightarrow \Re^n$ is a nonlinear vector function representing the nominal nonlinear state transition function of the system. However, modeling uncertainties always exist in practice. Thus, the actual system dynamics, may be represented by the following nonlinear state-space equations:

$$\begin{aligned} \dot{x}_r &= F_N(x_r, u) + \delta(x_r, u) \\ y_r &= x_r \end{aligned} \tag{5.2}$$

where $x_r \in \Re^n$ is the state vector of the actual system and $\delta : \Re^n \times \Re^m \rightarrow \Re^n$ is an unknown continuous nonlinear vector function representing the modeling uncertainties as a nonlinear

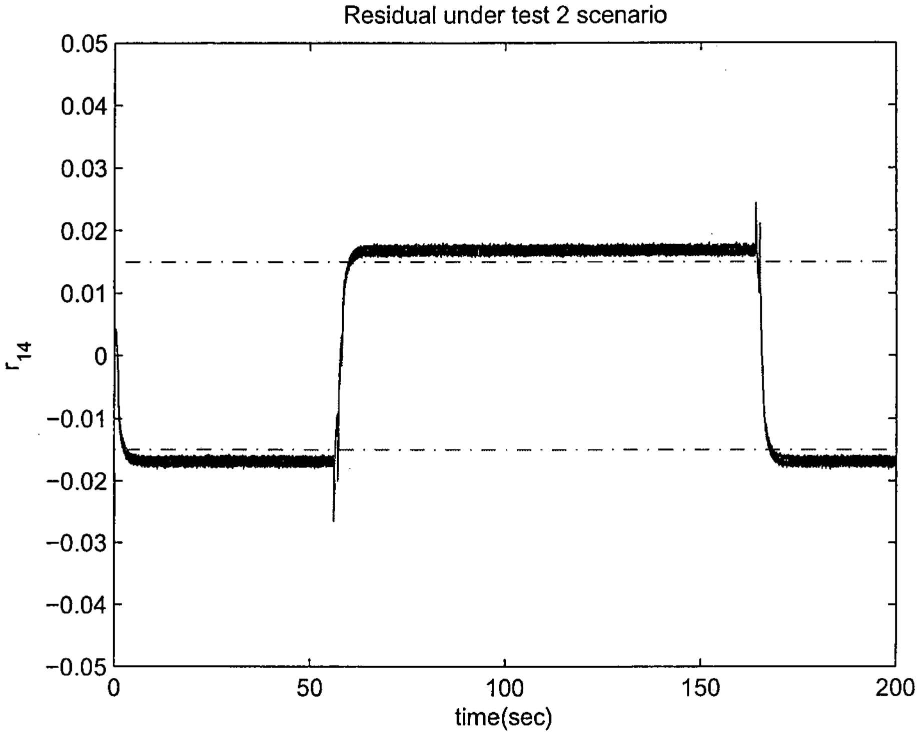


Figure 5.17: **Uncompensated residual r_{24} in the presence of modeling uncertainties (Test 2 scenario)**

function of system states and inputs.

By defining the effect of uncertainties on system states as the difference between the states of the actual system and the states of the nominal model given the same control input signals, we have:

$$x_{ud} = x_r - x_N \tag{5.3}$$

where x_{ud} is the effect of uncertainties on system states. Using full state measurement assumption and equation 5.3 we get:

$$y_{ud} = y_r - y_N = x_r - x_N = x_{ud} \tag{5.4}$$

where $y_r \in \Re^n$ are the actual measurements of the system, $y_N \in \Re^n$ are the outputs of the nominal model of the system and y_{ud} represents the effect of uncertainties on system outputs. Taking the time derivative of both sides of the equality in equation 5.3 and using equations 5.1 and 5.2, we have:

$$\begin{aligned} \dot{x}_{ud} &= \dot{x}_r - \dot{x}_N \\ &= F_N(x_r, u) + \delta(x_r, u) - F_N(x_N, u) \end{aligned} \tag{5.5}$$

Replacing x_N from equality $x_N = x_r - x_{ud}$ into equation 5.5, and using equation 5.4 we have:

$$\begin{aligned} \dot{x}_{ud} &= F_N(x_r, u) - F_N(x_r - x_{ud}, u) + \delta(x_r, u) \\ y_{ud} &= x_{ud} \end{aligned} \tag{5.6}$$

Equation 5.6 is a nonlinear state-space representation of a dynamical system with x_{ud} as states and x_r, u as inputs. Thus, equation 5.6 can be considered as the mathematical dynamic representation of the effect of uncertainties on the system states. Equation 5.6 can be rewritten in the following compact form:

$$\begin{aligned} \dot{x}_{ud} &= \psi(x_{ud}, x_r, u) \\ y_{ud} &= x_{ud} \end{aligned} \tag{5.7}$$

where $\psi : \Re^n \times \Re^m \rightarrow \Re^n$ is an unknown nonlinear function due to the unknown nature of the function $\delta(.,.)$.

5.4 Neural Networks in the Hybrid FDI Framework

Careful investigation of Figure 5.2 reveals that from signal flow point of view, the effect of modeling uncertainties will be propagated to the residual generation process of the geometric FDI scheme through the actual system outputs or measurements, represented by the outputs of the perturbed model. So, in order to avoid false alarms, we need to cancel out the effect of modeling uncertainties on the system outputs and provide the FDI subsystem with clean/compensated outputs.

In order to generate a set of clean/compensated outputs for the FDI subsystem, we have to subtract the effect of modeling uncertainties from system measurements. However, as mentioned previously, the unknown nature of the effects of modeling uncertainties does not allow the exact quantification of those effects. In other words, it is practically impossible to precisely and formally describe modeling uncertainties using differential or algebraic equations for the purpose of incorporating them in the design of model-based FDI algorithms. Thus, we need to approximate the unknown effects of modeling uncertainties in one way or another and then subtract the estimated effects from system output measurements.

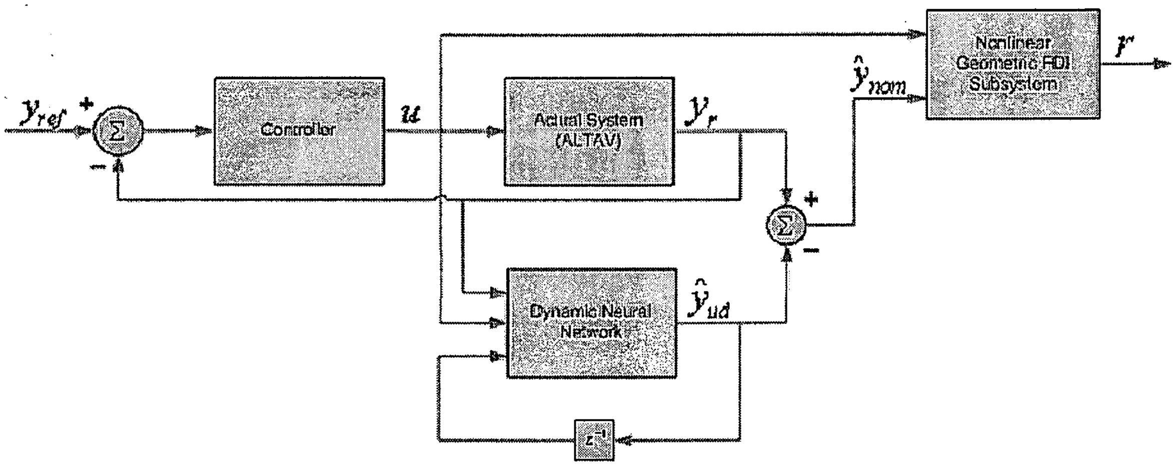


Figure 5.18: **Structure of the hybrid robust FDI scheme**

It is well-known that dynamic neural networks (DNN) are capable of approximating any nonlinear dynamic system provided sufficient number of layers and neurons are selected.

Thus, we will exploit DNN to approximate the dynamics of the modeling uncertainties, given in equation 5.7, to an arbitrary level of accuracy. The neural network will learn the dynamics of the modeling uncertainties through a learning/trainig process. Thus, the trained neural network is used in the on-line operation of the ALTAV, where the same control signals, generated by the ALTAV controller, will be applied to the inputs of the neural network. The block-diagram representation of the proposed robust FDI scheme with a trained dynamic neural network as the estimator of the modeling uncertainties effects is provided in Figure 5.18.

The data used for DNN training corresponds to the non-faulty scenario and the parameters of the DNN are adjusted such that the network can compensate the undesirable effects of the modeling uncertainties on system measurements that are fed to the model-based FDI residual generator. The block-diagram representation of the DNN training process is depicted in Figure 5.19. When the training is completed, the parameters of the network will be fixed or frozen and saved to be used in the on-line operation shown in Figure 5.18. The identification of a nonlinear dynamic system using neural networks will be discussed in the next section.

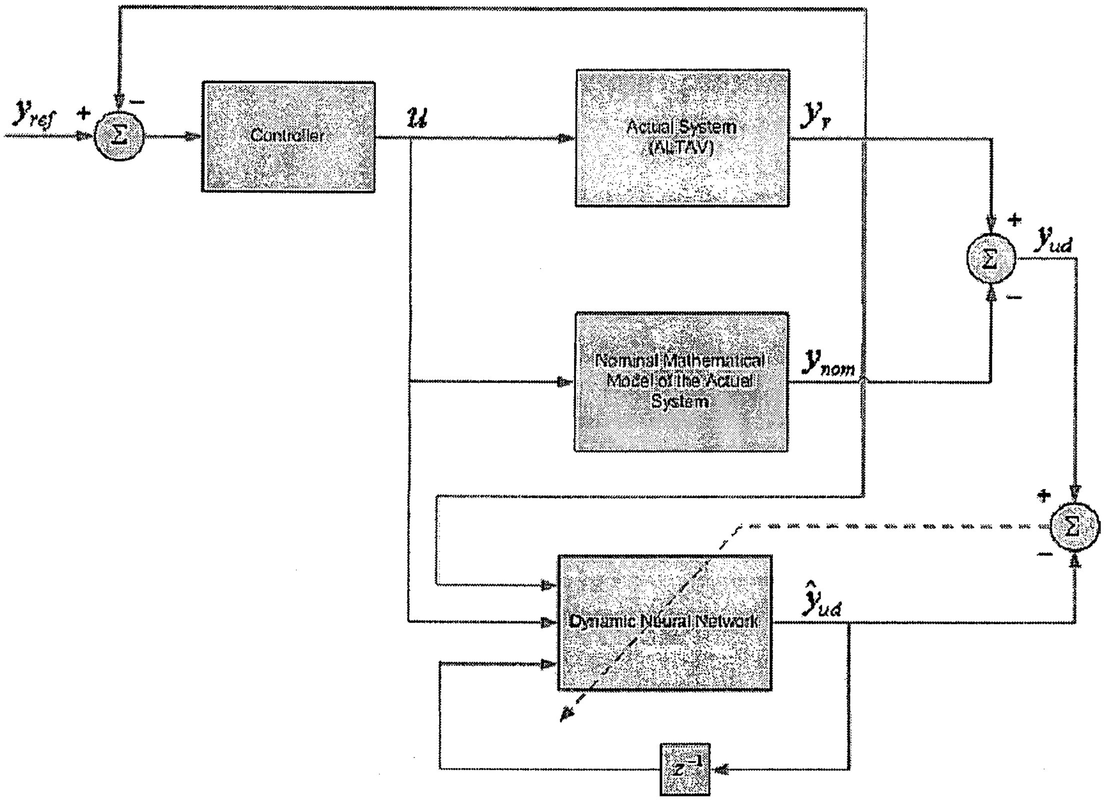


Figure 5.19: **Structure of the DNN training scheme for identification of modeling uncertainties**

5.5 Identification of Modeling Uncertainties using Dynamic Neural Networks

As mentioned previously, neural networks are very powerful tools in estimating unknown nonlinear functions. However, the most original representations of the neural networks do not incorporate any dynamics in the network architecture. Thus, they can not be employed to approximate nonlinear dynamic systems represented by nonlinear state-space equations such as the one in equation 5.7. Consequently, we require neural networks with dynamic elements incorporated into their architecture that are capable of representing nonlinear dynamic systems or equations. In general, such neural network architectures are called *dynamic neural networks*(DNN).

There is a large amount of publications on addressing the problem of introducing dynamics into the architecture of neural networks. As a result, different DNN architectures have been proposed and developed in the literature. In the next section, we will introduce some of these architectures utilized primarily for nonlinear dynamic system identification and eventually select one that suits our robust FDI scheme the best.

5.5.1 Literature Review of Dynamic Neural Networks for System Identification

A large body of literature has been dedicated to the identification of a general class of nonlinear dynamic system using neural networks. These work are motivated from and justified by the following three important features of neural networks [71], namely (i) their nonlinear characteristics that make them suitable for dealing with nonlinear systems, (ii) their parallel and pipeline processing characteristics that allow them to perform different tasks more efficiently, and (iii) their learning characteristics that are ideal for adapting to different environmental conditions.

Among the work in the literature one may classify the schemes into four main categories. The first category ([72], [60]) utilize tapped delay lines along with a static network to generate dynamics in the structure. Basically, in this method delay elements are used to introduce delayed inputs and outputs that are then fed to a static network as the regressor vector. The network then performs a static nonlinear map on this regressor vector so that the desired output is obtained.

The second category is the recurrent neural networks. In this approach by using a recurrent structure a dynamic input-output representation is constructed. This method has been investigated in [73], [74], [75]. Especially, in [73] it is shown analytically that the proposed recurrent neural network is capable of identifying any nonlinear dynamic system provided that the initial states of the network are chosen appropriately with respect to the initial conditions of the system.

The third category which has not been studied as extensively as the first two is the embedded dynamic neural networks. Embedded dynamic neural networks are constructed by utilizing dynamic neurons. The dynamic neuron structure is different from that of static neuron in the sense that in the former one or more dynamic elements are utilized to obtain a specific dynamical input-output map. Several dynamic neuron structures have been reported in references [76], [77], [78],[79]. In [76] a spatio-temporal neuron was introduced in which the conventional weight multiplication operation was replaced by a linear filtering (an all zero filter) operation. Gamma model was utilized in [77] for identification of nonlinear systems. The structure of the Gamma model is similar to the tapped delay line structure but instead of using simple shift elements in the line, a first order linear filter is utilized to generate a dynamic input-output map. In [78] a dynamic neural processor was proposed based on the physiological fact that the neural activities are dependent upon the interaction of antagonistic neural sub-populations, excitatory and inhibitory activities. The well known Time Delay Neural Network (TDNN) was introduced in [80] for phoneme recognition. In TDNN each weight is associated with a delay. The adaptive version of TDNN was utilized in [79] for identifying two classes of nonlinear systems referred to as "the first" and "the fourth" class of nonlinear systems [60].

In [71], a neuro-dynamic identifier is developed by using a dynamic neural network with embedded linear adaptive filters. It is shown that by utilizing a specific type of dynamic neurons the simplest proposed structure is capable of representing the input-output map of a class of nonlinear systems denoted as "the first class of nonlinear systems" in [60]. The proposed structure was also modified and generalized to represent the input-output map of the other three classes of nonlinear systems. For the DNN structure proposed in [71], learning takes place by adapting the embedded linear filter parameters as well as the neural network weights. The adaptation of the linear filter parameters will not only increase the computational complexity of the DNN but also the risk of instability of the filter during the learning process.

The fourth class of DNN structures that is proposed in [81] [82] consists of a feed-forward static neural network architecture cascadede/followed by a fixed stable linear filter. The authors have developed the proposed DNN for stable identification of unknown nonlinear systems. In the proposed structure, the only adaptive parameters of the DNN during the training/learning process are the neural weights and the parameters of the stable linear filter remain unchanged. Furthermore, a novel approach has also been proposed for the weight update mechanism based on the modification of the well-known back-propagation (BP) algorithm. Figure 5.20 shows the original structure of the proposed DNN [82]. The original DNN structure proposed in [81] [82] uses continuous-time filter in the architecture. In our proposed robust FDI scheme depicted in Figure 5.18, we have employed the same DNN architecture with the slight modification of using discrete-time rather than of a continuous-time stable linear filter. Correspondingly, the weight update laws have also been slightly modified. The two motivations behind the selection of such DNN architecture are (i) the existence of a formal and analytical stability proof for the overall identification system using Lyapunov's direct method, and (ii)less computational and analytical complexity of the training process and weight update mechanism due to the fact that the filter parameters are not getting updated.

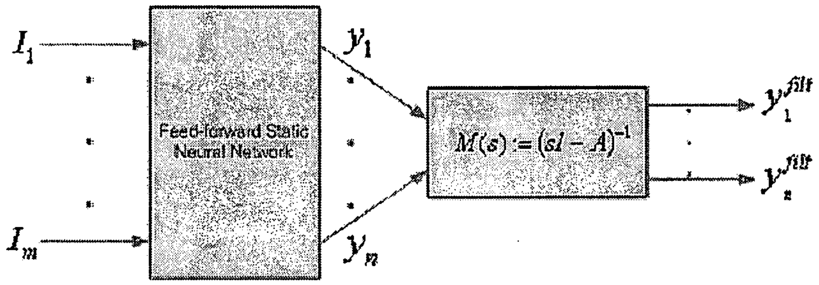


Figure 5.20: **Structure of the selected DNN**

5.5.2 Application of DNN to Identification of ALTAV Modeling Uncertainties

In order to make the FDI subsystem with respect robust to modeling uncertainties, we require to compensate only the impact of uncertainties on those outputs of the system that are used as inputs to the FDI subsystem. Although we have twelve outputs for the ALTAV platform, however, only six of them, namely $\dot{x}, \gamma, \phi, \dot{\theta}, \dot{\gamma}, \dot{\phi}$, are utilized as inputs to the nonlinear geometric FDI subsystem. On the other hand, we observed previously that the introduced parameter uncertainties given in Table 5.2 only force the 5th and the 6th residuals, namely r_{13}, r_{24}, to go out of threshold bounds. However, careful investigation of the observer equations given in equation 4.47 reveals that the observers generating the residuals r_{13}, r_{24}, use only four of the ALTAV outputs, namely $\phi, \dot{\theta}, \dot{\gamma}\dot{\phi}$. So, we need to identify and consequently compensate the effect of modeling uncertainties only on these four outputs.

In order to identify the effect of modeling uncertainties on the above-mentioned four outputs of interest using DNN, we need to specify first the inputs to the DNN structure. Analysis of the ALTAV dynamics given in equation 4.35 reveals that the two outputs $\gamma, \dot{\gamma}$ are decoupled from the two other outputs $\phi, \dot{\phi}$, and each pair corresponds to a separate second-order nonlinear differential equation. Therefore, we require two DNN structures each identifying the modeling uncertainties effects on a separate pair of outputs. The structure of these two DNNs are shown in Figure 5.21.

5.5.3 DNN Identifier Equations and Update Laws

Consider the two DNN structures in Figure 5.21. We define the state vector for each DNN as follows:

$$x_{ud}^{\gamma} = [\hat{\gamma}_{ud} \quad \hat{\dot{\gamma}}_{ud}]^{\top} \tag{5.8}$$

$$x_{ud}^{\phi} = [\hat{\phi}_{ud} \quad \hat{\dot{\phi}}_{ud}]^{\top} \tag{5.9}$$

Then, by defining the input vectors of the two DNNs as:

$$\bar{x}^{\gamma}(t) = [x_r^{\gamma}(t) \quad F_1(t) \quad F_3(t) \quad x_{ud}^{\gamma}(t-1)]^{\top} \tag{5.10}$$

$$\bar{x}^{\phi}(t) = [x_r^{\phi}(t) \quad F_2(t) \quad F_4(t) \quad x_{ud}^{\phi}(t-1)]^{\top} \tag{5.11}$$

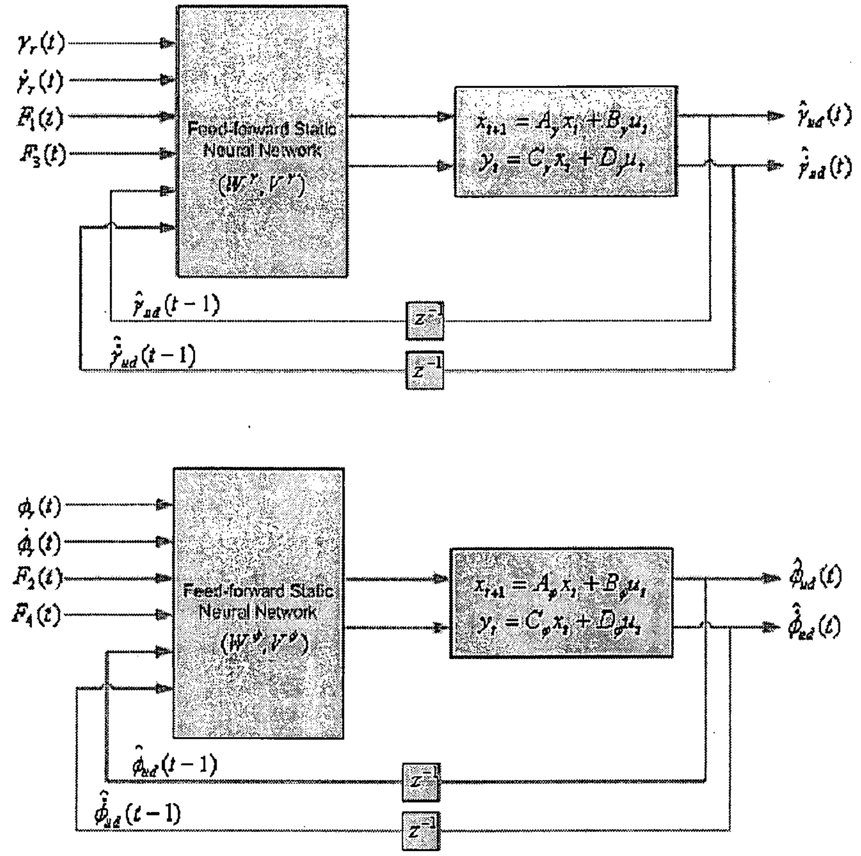

$\gamma_r(t)$
$\dot{\gamma}_r(t)$
$F_1(t)$
$F_3(t)$
Feed-forward Static Neural Network
(W^{γ}, V^{γ})
$x_{t+1} = A_{\gamma}x_t + B_{\gamma}u_t$
$y_t = C_{\gamma}x_t + D_{\gamma}\mu_t$
$\hat{\gamma}_{ud}(t)$
$\hat{\dot{\gamma}}_{ud}(t)$
$\hat{\gamma}_{ud}(t-1)$
z^{-1}
$\hat{\dot{\gamma}}_{ud}(t-1)$
z^{-1}
$\phi_r(t)$
$\dot{\phi}_r(t)$
$F_2(t)$
$F_4(t)$
Feed-forward Static Neural Network
(W^{ϕ}, V^{ϕ})
$x_{t+1} = A_{\phi}x_t + B_{\phi}u_t$
$y_t = C_{\phi}x_t + D_{\phi}\mu_t$
$\hat{\phi}_{ud}(t)$
$\hat{\dot{\phi}}_{ud}(t)$
$\hat{\phi}_{ud}(t-1)$
z^{-1}
$\hat{\dot{\phi}}_{ud}(t-1)$
z^{-1}

Figure 5.21: **Two DNNs each identifying modeling uncertainties for a pair of output variables**

we will have the following state-space representation for the dynamics of the two DNNs:

$$x^{\gamma}_{ud}(t) = A_{\gamma}x^{\gamma}_{ud}(t-1) + G^{\gamma}(x^{\gamma}_{ud}(t-1), x^{\gamma}_{r}(t), F_1, F_3) \tag{5.12}$$

$$x^{\phi}_{ud}(t) = A_{\phi}x^{\phi}_{ud}(t-1) + G^{\phi}(x^{\phi}_{ud}(t-1), x^{\phi}_{r}(t), F_2, F_4) \tag{5.13}$$

where A_{γ}, A_{ϕ} are the Hurwitz matrices of the stable linear filters of the two DNNs and the nonlinear functions G^{γ}, G^{ϕ} are the nonlinear mappings of the feed-forward neural networks in the structure of each DNN, and are given by:

$$G^{\gamma}(\bar{x}^{\gamma}) = W^{\gamma}\sigma(V^{\gamma}\bar{x}^{\gamma}) \tag{5.14}$$

$$G^{\phi}(\bar{x}^{\phi}) = W^{\phi}\sigma(V^{\phi}\bar{x}^{\phi}) \tag{5.15}$$

where W, V are the weights of the hidden and output layers of the feed-forward network, respectively, and $\sigma(.)$ is the transfer/activation function of the hidden neurons that is usually considered as a sigmoidal function:

$$\sigma_i(V_i\bar{x}) = \frac{2}{1 + exp[-2V_i\bar{x}]} - 1 \tag{5.16}$$

where V_i is the ith row of V, and $\sigma_i(V_i\bar{x})$ is the ith element of the vector $\sigma(V\bar{x})$.

In the proposed DNN structure, only the neural weights of the static feed-forward maps, namely $W^{\gamma}, V^{\gamma}, W^{\phi}, V^{\phi}$ are getting updated. The update of the weights is carried out iteratively using a modified gradient descent algorithm [82] as given in equations 5.17. This general formulation of the update laws will essentially remain the same for the output weights V, so they have only been given for the hidden layer weights, namely W. However, the derivation of partial derivatives will be different for the hidden and output weights, which will result in different final update laws, as provided in equations 5.19.

$$W^{\gamma}(t+1) = W^{\gamma}(t) - \eta_w^{\gamma}(\frac{\partial J^{\gamma}}{\partial W^{\gamma}(t)}) - \rho_w^{\gamma}||\tilde{x}_{\gamma}||W^{\gamma}(t) \tag{5.17}$$

$$W^{\phi}(t+1) = W^{\phi}(t) - \eta_w^{\phi}(\frac{\partial J^{\phi}}{\partial W^{\phi}(t)}) - \rho_w^{\phi}||\tilde{x}_{\phi}||W^{\phi}(t) \tag{5.18}$$

The first term in the above equations is the back-propagation (BP) term and the second term is the e-modification term. Furthermore, $\eta_w^{\gamma}, \eta_w^{\phi} > 0$ are the learning rates, $\rho_w^{\gamma}, \rho_w^{\phi}$ are small positive numbers, $J^{\gamma} = \frac{1}{2}(\tilde{x}_{\gamma}^{\top}\tilde{x}_{\gamma})$ and $J^{\phi} = \frac{1}{2}(\tilde{x}_{\phi}^{\top}\tilde{x}_{\phi})$ are the objective functions associated to each DNN, and $\tilde{x}_{\gamma}, \tilde{x}_{\phi}$ are the identification errors of each DNN identifier. Calculation of the partial derivatives will result in the following update laws for the all the weights of the two identifiers:

$$W^{\gamma}(t+1) = W^{\gamma}(t) - \eta_w^{\gamma}(\tilde{x}_{\gamma}^{\top}(I - A_{\gamma})^{-1})^{\top}(\sigma(V^{\gamma}\bar{x}^{\gamma}))^{\top} + \rho_w^{\gamma}||\tilde{x}_{\gamma}||W^{\gamma}(t) \tag{5.19}$$

$$V^{\gamma}(t+1) = V^{\gamma}(t) - \eta_v^{\gamma}(\tilde{x}_{\gamma}^{\top}(I - A_{\gamma})^{-1}W^{\gamma}(I - \Lambda(V^{\gamma}\bar{x}^{\gamma})))^{\top}\bar{x}^{\top} + \rho_v^{\gamma}||\tilde{x}_{\gamma}||V^{\gamma}(t) \tag{5.20}$$

$$W^{\phi}(t+1) = W^{\phi}(t) - \eta_w^{\phi}(\tilde{x}_{\phi}^{\top}(I - A^{\phi})^{-1})^{\top}(\sigma(V^{\phi}\bar{x}^{\phi}))^{\top} + \rho_w^{\phi}||\tilde{x}_{\phi}||W^{\phi}(t) \tag{5.21}$$

$$V^{\phi}(t+1) = V^{\phi}(t) - \eta_v^{\phi}(\tilde{x}_{\phi}^{\top}(I - A^{\phi})^{-1}W^{\phi}(I - \Lambda(V^{\phi}\bar{x}^{\phi})))^{\top}\bar{x}^{\top} + \rho_v^{\phi}||\tilde{x}_{\phi}||V^{\phi}(t) \tag{5.22}$$

5.6 Simulation Results

In order to evaluate the effectiveness of the proposed robust FDI scheme for the ALTAV platform, we first require to identify the dynamics of modeling uncertainties through a training process depicted in Figure 5.19. Due to inaccessibility of an actual ALTAV we had to represent the actual ALTAV with the nominal model of the ALTAV, given in equation 4.35 but with the parameters of the model perturbed from their nominal values, as given in Tables 5.1 and 5.2. Furthermore, a circular reference trajectory was used for the ALTAV closed-loop operation.

As shown in Figure 5.21, two DNN identifiers were used each consisting of a one-hidden layer feed-forward neural networks with 6 neurons in the input and 8 neurons in the hidden layers, respectively. The values of the other parameters of the two DNN identifiers such as the learning rates and state transition matrix of the discrete-time filter are given in below:

1) DNN parameter values for identifying the effect of modeling uncertainties on $\gamma, \dot{\gamma}$:

$$\begin{aligned}
\eta_w^\gamma &= \eta_v^\gamma = 10^{-4} \\
\rho_w^\gamma &= \rho_v^\gamma = 10^{-6} \\
A_\gamma &= 0.99 I_{2\times 2}
\end{aligned}$$

2) DNN parameter values for identifying the effect of modeling uncertainties on $\phi, \dot{\phi}$:

$$\begin{aligned}
\eta_w^\phi &= \eta_v^\phi = 10^{-5} \\
\rho_w^\phi &= \rho_v^\phi = 10^{-7} \\
A_\phi &= 0.99 I_{2\times 2}
\end{aligned}$$

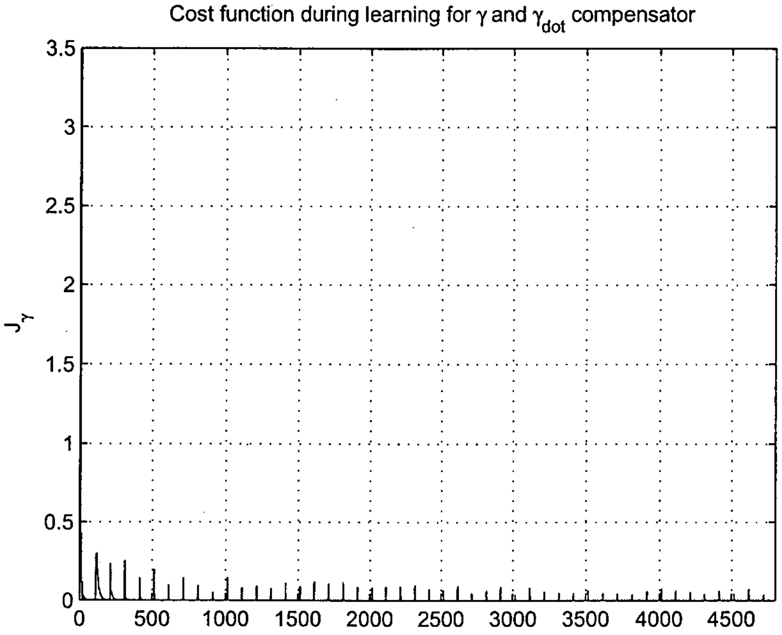


Figure 5.22: Cost of training for γ

The training was performed for 4800(sec) of closed-loop operation of the ALTAV, however, the learning rates were set to zero, i.e. weight adaptation was stopped and the weights were frozen, at $t_{stop} = 4300(sec)$. Figures 5.22 and 5.23 show the cost functions for the two DNNs. As can be seen from the figures, the cost function is very well decreasing over time. More importantly, even after the weight update was stopped ($t_{stop} < t < 4800$), the cost functions remain at the same level as before the stop time ($t < t_{stop}$), which is an indication of the success of the training process of DNNs in learning the modeling uncertainties effects.

After the termination of the training process, the freezed weights of the DNNs are saved and used in the on-line closed-loop operation of the ALTAV with robust FDI scheme depicted in Figure 5.18. Then, the system shown in Figure 5.18 was ran for 200(sec) with the original rectangular reference trajectory under the presence of modeling uncertainties and absence of faults. Figures 5.25, 5.26, 5.28, 5.29, 5.31, and 5.32 show the residuals r_{13}, r_{24} generated by

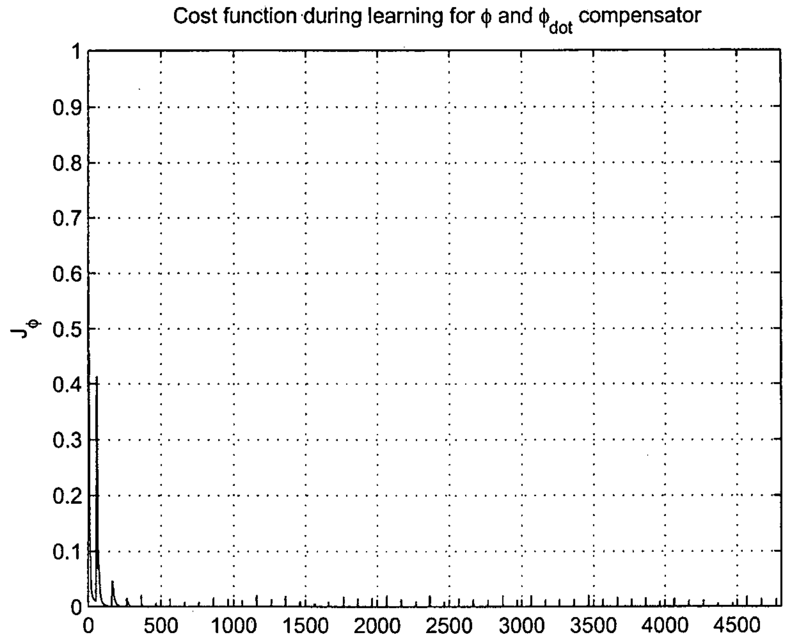


Figure 5.23: Cost of training for ϕ

the robust FDI scheme, respectively. It can be seen in the figures that even in the presence of modeling uncertainties, the two residuals stay within their threshold bounds except for a very small negligible period of time for r_{24}. Thus, no false alarms are generated by the robust FDI scheme as opposed to the results of the non-robust FDI scheme, presented in Figures ??, ??. Therefore, the simulation results confirm the reliability of the performance of the proposed robust FDI scheme.

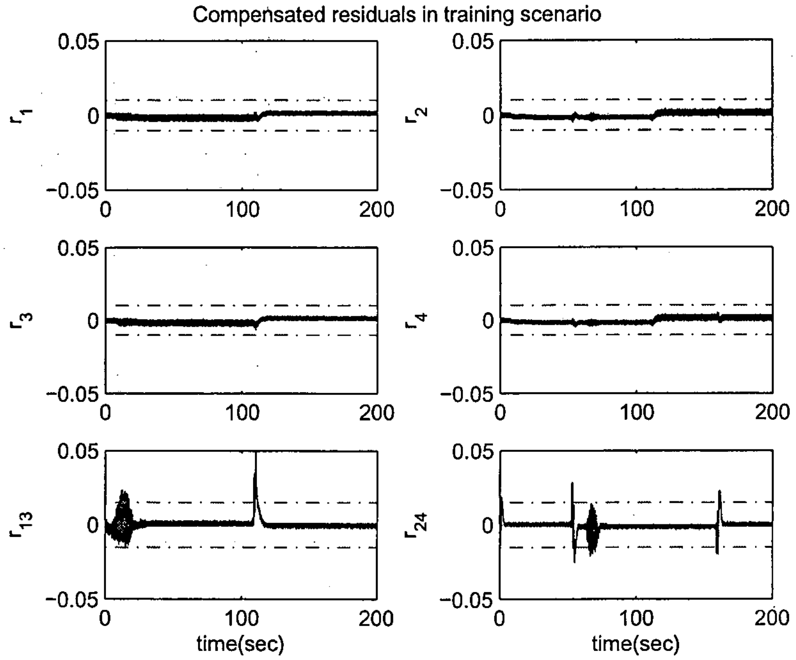


Figure 5.24: All compensated residuals in training scenario

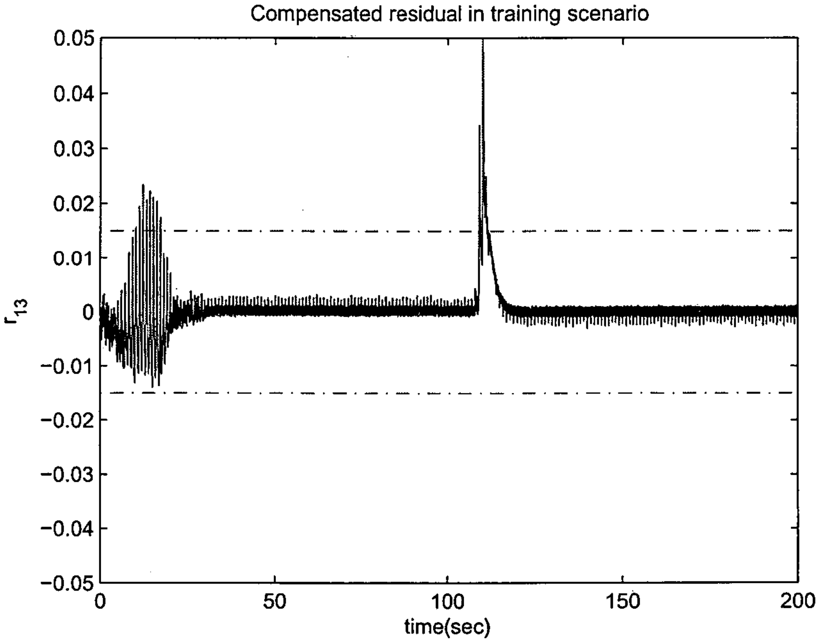


Figure 5.25: Compensated residual r_{13} in training scenario

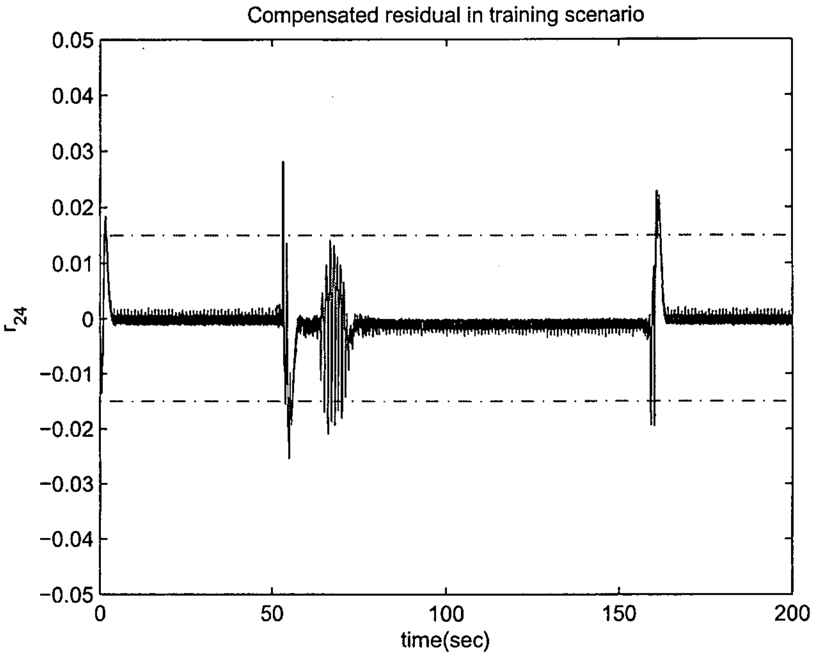


Figure 5.26: Compensated residual r_{24} in training scenario

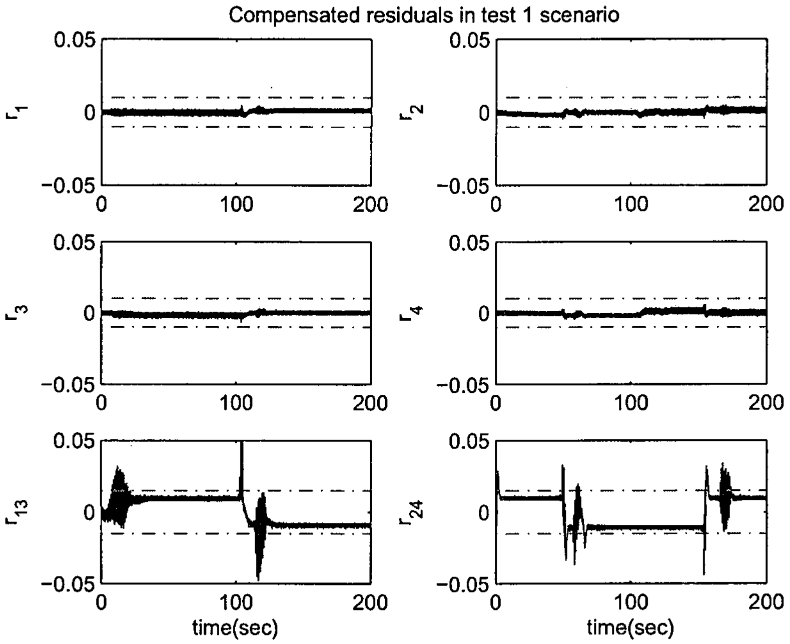


Figure 5.27: All compensated residuals in test 1 scenario

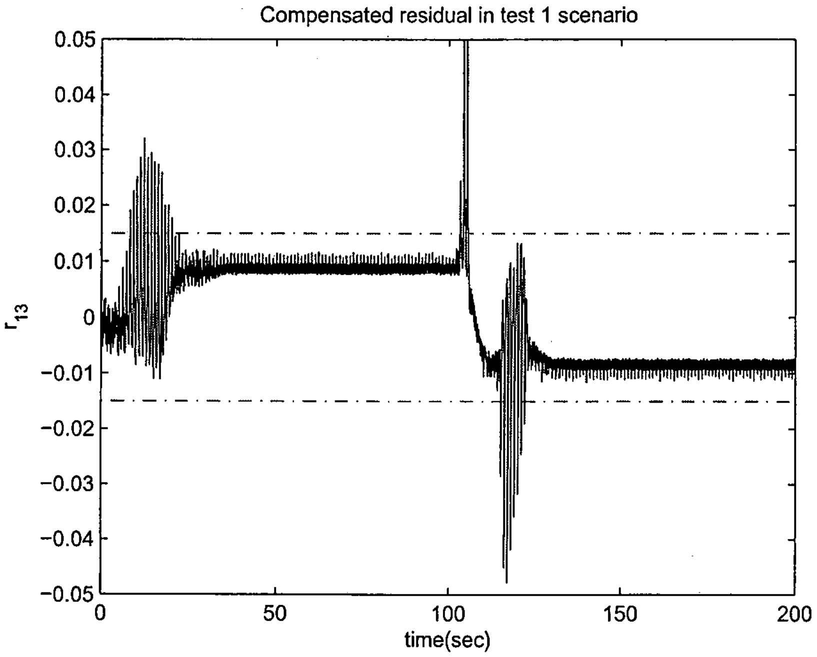


Figure 5.28: Compensated residual r_{13} in test 1 scenario

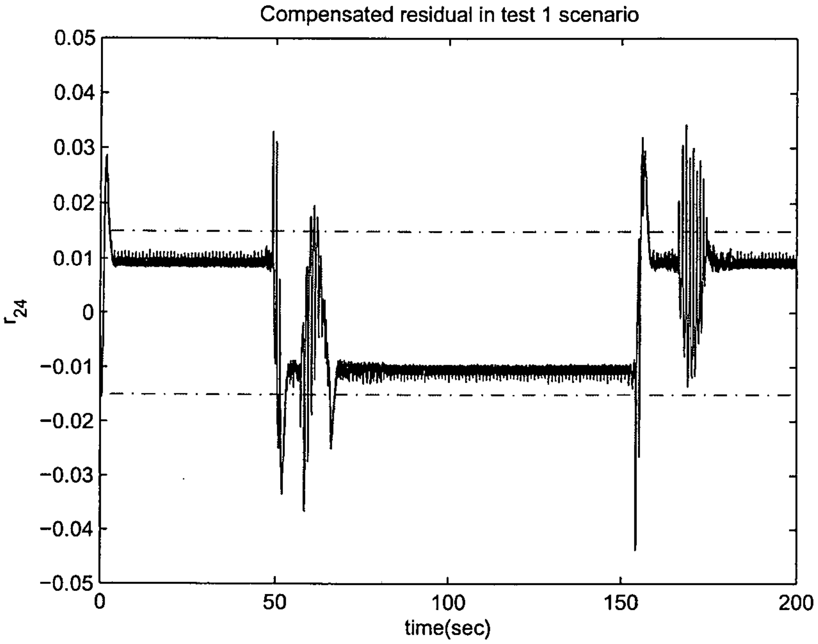


Figure 5.29: Compensated residual r_{24} in test 1 scenario

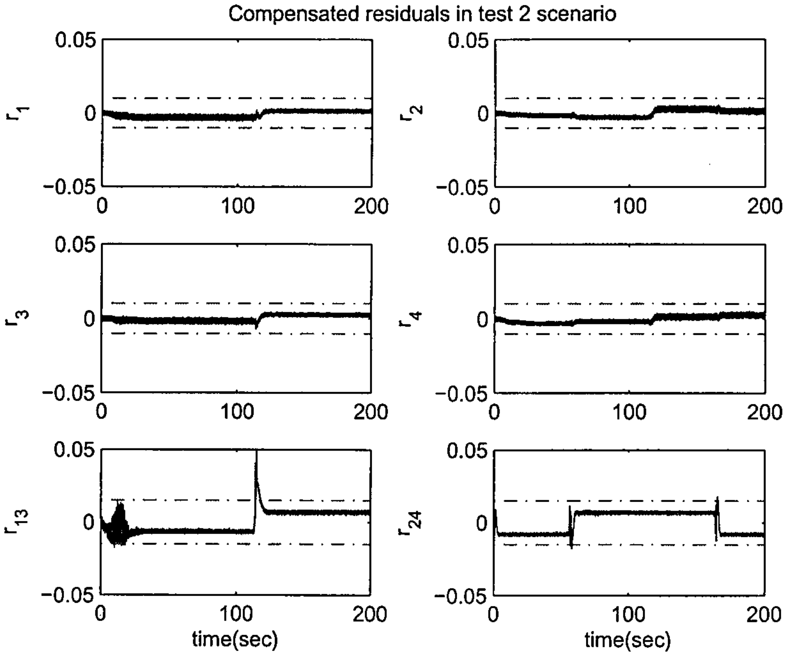


Figure 5.30: All compensated residuals in test 2 scenario

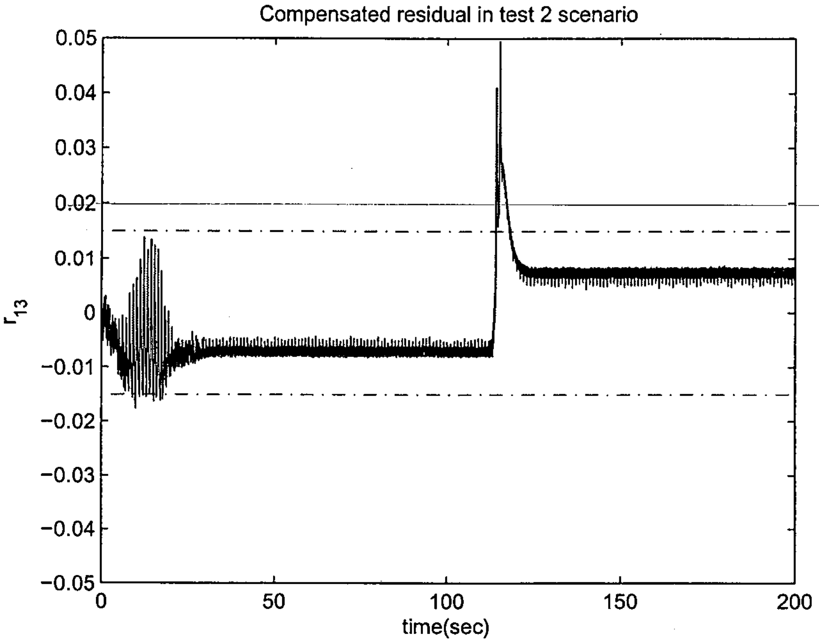


Figure 5.31: Compensated residual r_{13} in test 2 scenario

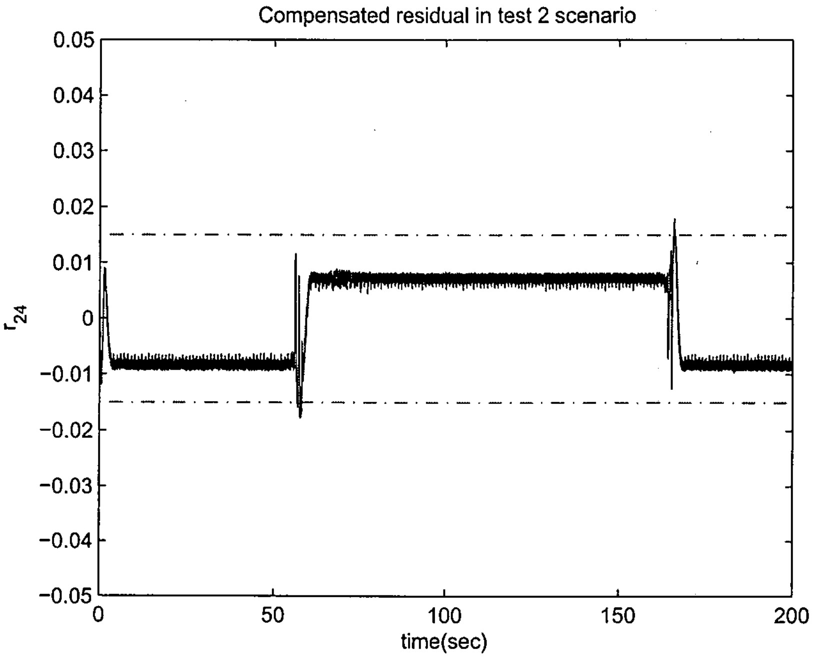


Figure 5.32: Compensated residual r_{24} in test 2 scenario

Chapter 6

Cooperative Rendezvous of UAVs

6.1 Rendezvous Cooperative Control

A number of papers ([33] [83] [84] [85] [5]) have investigated this problem. Their techniques will be reviewed here and will be applied to our scenarios.

6.1.1 Rendezvous Scenarios

Consider a scenario where a group of M unmanned air vehicles (UAVs) are required to transition through N known target locations. In the region of interest, there are a number of threats, some known *a priori*, others "pop up," or become known only when a UAV maneuvers into their proximity. Suppose that to maximize the probability that the mission will succeed, it is desirable to have multiple UAVs arrive on the boundary of each target's radar detection region simultaneously.

The scenario can be decomposed into several subproblems.

(i) Given M UAVs with N targets, assign each vehicle to a target such that each target has, if possible, multiple UAVs assigned to it, and such that preference is given to high-priority targets.

(ii) For each team of UAV's assigned to a single target, determine an estimated time over target(TOT) that ensures simultaneous intercept and that is feasible for each UAV on the team.

(iii) For each UAV with velocity constraints $v \in [V_{min}, V_{max}]$, determine a path (specified via waypoints), such that if the UAV were to fly along straight-line paths, it could complete the path in the specified TOT, while satisfying the given velocity constraints.

(iv) Transform each waypoint path into a feasible trajectory for the UAV. By feasible, we mean that the turning rate constraints and velocity constraints are not violated along the trajectory. Also, the trajectory should have the same TOT as the path specified by waypoints.

(v) Develop globally asymptotically stable controllers for each UAV, such that the UAVs track their specified trajectory.

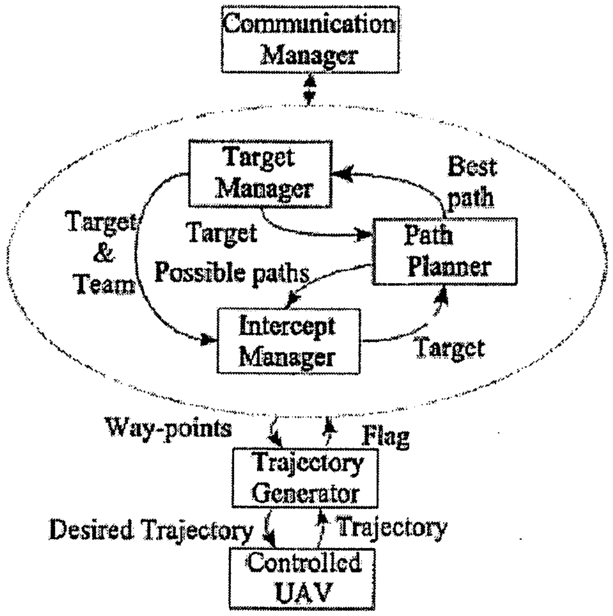


Figure 6.1: System architecture for a single UAV [5]

Please be noted that the solution proposed by Beard and McLain ([5] [83] [84] [85] [33]) assumes that individual UAVs fly at different, preassigned altitudes thereby ensuring collision avoidance.

In this project, in order to simplify the rendezvous problem and concentrate on the FDIR techniques for rendezvous UAV controls, we only consider that there are only a number of known threats, three UAVs and one target. That is, the all three vehicles are expected to arrive the target simultaneously in the presence of a number of known threats. As a result, the subproblems we need to solve in this project is (i) coordinated UAV intercept, (ii) path planning, (iii) feasible trajectory generation, and (iv) asymptotic trajectory following. A detailed schematic of the system architecture for a single UAV is shown in Figure 6.1. At the lowest level of the architecture is the physical UAV. In Figure 6.1, the target manager, path planning, and intercept manager work together to generate waypoint paths for the UAV. The path planner generates a specified number of paths from the specified UAV to a specific target. The path planner returns in-formation about each path, namely, the estimated fuel expenditure and the estimated threat exposure. Both the target manager and the intercept manager make calls to the path planner. The role of the intercept manager is to ensure that when the target manager assigns multiple UAVs to the same target, they arrive on the radar detection boundary of the target simultaneously.

The trajectory generation block in Figure 6.1 receives a set of waypoints, which specify the desired path of the UAV. The role of the trajectory generator is to generate a feasible time trajectory for the UAV. By feasible, we mean that in the absence of disturbances and modeling errors, an input trajectory causes the UAV fly the trajectory without violating its velocity and heading rate constraints.

The communication manager shown in Figure 6.1 facilitates communication between different UAVs. Each UAV implements a separate target manager, intercept manager, and path

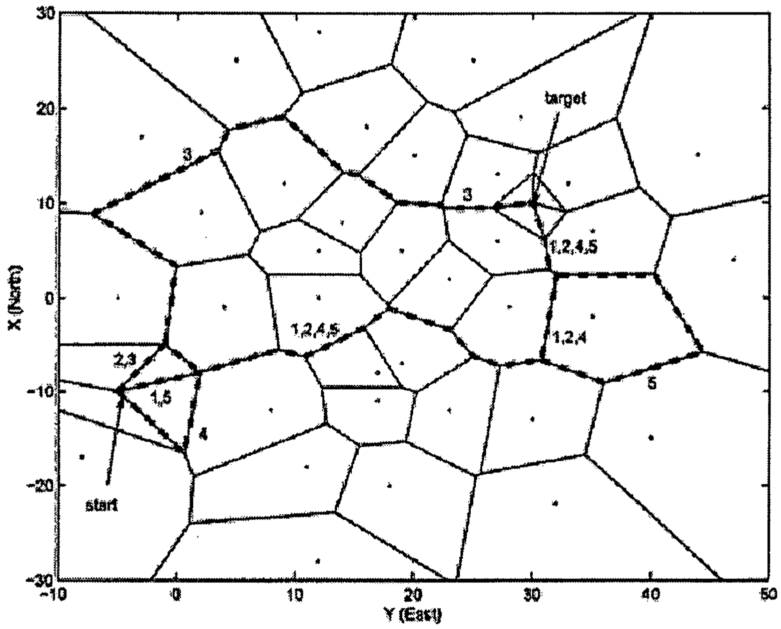


Figure 6.2: Threat-based Voronoi diagram [5]

planner. Therefore, the decisions reached by these functional blocks must be synchronized among the different UAVs.

As we mentioned previously, in this project, all three UAVs are expected to arrive one target simultaneously, so target manager will not be investigated, and path planner and intercept manager techniques will be reviewed in the next subsections.

6.2 Path Planner

The output of the path planner is a set of waypoints and commanded velocity for each vehicle. Paths from the initial vehicle location to the target location are derived from a k-best paths graph search of a Voronoi diagram that is constructed from the known threat locations [86]. Creating the Voronoi diagram entails partitioning the region of interest with n threats into n convex polygons or cells. Each cell contains exactly one threat, and every location within a given cell is closer to its associated threat than to any other threat. By using threat locations to create the diagram, the resulting Voronoi polygon edges form a set of lines that are equidistant from the closest threats, thereby maximizing their distance from the closest threats. The initial and target locations are also contained within cells and are connected to the nodes forming the cell. Figure 6.2 shows a Voronoi diagram created for a set of threats, UAV location, and target location.

Each edge of the Voronoi diagram is assigned two costs: a threat cost and a length cost. Threat costs are based on a UAV's exposure to a radar located at the threat. Assuming that the UAV radar signature is uniform in all directions and is proportional to $1/d^4$ (where d is the distance from the UAV to the threat), the threat cost for traveling along an edge is inversely proportional to the distance to the fourth power. In this work, the threat cost was calculated at three points along each edge: $L_i/6$, $L_i/2$, and $5L_i/6$, where L_i is the length of edge i. The threat cost associated with the ith edge is given by the expression

$$J_{threat,i} = \frac{\alpha L_i}{3} \sum_{j=1}^{N} \left(\frac{1}{d_{1/6,i,j}^4} + \frac{1}{d_{1/2,i,j}^4} + \frac{1}{d_{5/6,i,j}^4} \right) \tag{6.1}$$

where N is the total number of threats, $d_{1/2,i,j}$ is the distance from the 1/2 point on the ith edge to the jth threat, and α is the constant scale factor.

To include the path length as part of the cost, the length cost associated with each edge is

$$J_{length,i} = L_i \tag{6.2}$$

The total cost for traveling along an edge comes from a weighted sum of the threat and length costs

$$J_i = \kappa J_{length,i} + (1 - \kappa) J_{threat,i} \tag{6.3}$$

The choice of κ between 0 and 1 gives the designer flexibility to place weight on exposure to threats or fuel expenditure depending on the particular mission scenario. For this work, a value of 0.25 was found to produce paths that were balanced in terms of threat avoidance and path length.

With the cost determined for each of the Voronoi edges, the Voronoi diagram is searched to find the set of lowest-cost candidate paths between the initial UAV location and the location of the target. The graph search is carried out using a k-best paths algorithm.

6.3 Intercept Manager

In the provided architecture, the vehicles are required to prosecute their target simultaneously to enhance mission effectiveness. The challenge is to determine the best TOT specification for the team in light of the threat scenario and dynamic capabilities of the vehicles. The output of the intercept manager is a velocity and a set of waypoints for each vehicle. Once candidate paths have been determined by searching the Voronoi graph, it remains to select which path each UAV should fly. Path selection should result in minimal collective threat exposure for the team and simultaneous arrival of team members at their target.

Candidate paths to the target for UAV i are parameterized by the waypoints ξ_i and the UAV forward speed V_i. TOT for each UAV is function of the speed of the UAV along the selected path: $\text{TOT}_i = f(\xi_i, V_i)$. Similarly, the cost (threat and fuel) to travel along any path to the target is a function of the path and the speed: $\mathcal{J}_i = g(\xi_i, V_i)$. To choose the best TOT for the team in a cooperative manner, some information regarding the candidate paths for the UAVs must be exchanged. Rather than passing ξ_i and V_i around among all of the UAVs, a more efficient parameterization of information, called the coordination function, is used. In this problem, the coordination function $\hat{\mathcal{J}}_i$ models the cost to UAV i of achieving a particular TOT: $\hat{\mathcal{J}}_i = h(\text{TOT})$. The coordination function is determined from the relations for TOT_i and $\hat{\mathcal{J}}_i$. Based on the candidate paths determined from the Voronoi diagram and feasible range of UAV air speeds, the feasible TOT range for each UAV on the team can also be determined.

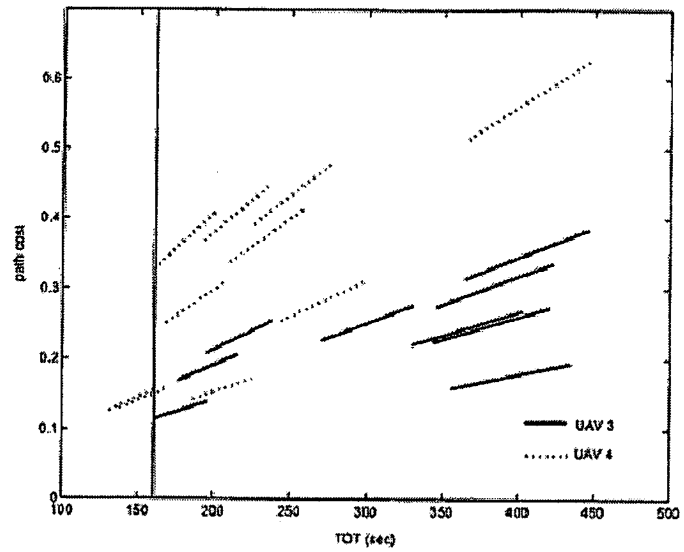


Figure 6.3: Coordinate functions for two UAVs [5]

The cost $\mathcal{J}_i$ is given by a linear combination of threat cost and fuel cost:

$$\mathcal{J}_i = (1-\kappa)\mathcal{J}_{threat,i} + \kappa\mathcal{J}_{fuel,i} \tag{6.4}$$

The threat cost for each edge of a path from a threat is represented by:

$$\mathcal{J}_{threat,i,j} = \frac{\rho\kappa^2}{V_i}\int_{s=0}^{1}\frac{1}{\|h-(w_{j-1}+s(w_j-w_{j-1}))\|^4}ds \tag{6.5}$$

where h represents the threat's position, and $s \in [0,1]$ parameterizes the straight-line path from w_{j-1} to w_j. A closed-form solution to this integral is given by

$$\begin{aligned}\mathcal{J}_{threat,i,j} = &\frac{\rho\kappa^2}{2V_i b^2\|w_j-w_{j-1}\|^3}\left[\frac{1-a}{(1-a)^2+b^2}+\frac{1}{b}\tan^{-1}\left(\frac{1-a}{b}\right)\right.\\ &\left.+\frac{a}{a^2+b^2}+\frac{1}{b}\tan^{-1}\left(\frac{a}{b}\right)\right]\end{aligned} \tag{6.6}$$

where

$$a = \frac{(h-w_{j-1})^T(w_j-w_{j-1})}{\|w_j-w_{j-1}\|^2}$$

$$b = \frac{\|\left[\|w_j-w_{j-1}\|^2 I-(w_j-w_{j-1})(w_j-w_{j-1})^T\right](h-w_{j-1})\|}{\|w_j-w_{j-1}\|^3}$$

The fuel cost for traversing an edge is calculated based on the assumption that fuel usage is proportional to aerodynamic drag force, which is proportional to velocity squared. Accordingly, the fuel required to traverse an edge of a waypoint path from w_{j-1} to w_j is given by

$$\mathcal{J}_{fuel,i,j} = \int_{t_{j-1}}^{t_j}\dot{f}dt = c_{fuel}V_i^2(t-t_{j-1}) = c_{fuel}V_i\|w_j-w_{j-1}\| \tag{6.7}$$

where $c_{fuel} > 0$ is a constant.

Coordination function information for two UAVs composing a team is shown in Figure 6.3. Note that each candidate path produces a line segment in the plot.

Coordinate function and feasible TOT range information are exchanged among vehicles to enable cooperative path planning. At the team level, the cooperative planning problem is simplified to finding TOT* that minimizes the collective threat exposure of the team

$$\mathrm{TOT}^* = \arg\min \sum_{i=1}^{n} \hat{\mathcal{J}}_i$$

From Figure 6.3, it can be seen that this optimization results in a team-optimal value of TOT*=160.5s. TOT is called the coordination variable. By requiring that individual UAV match the team-optimal value of the coordination variable TOT*, cooperation is ensured. Once TOT* has been determined, each UAV must determine which of the candidate paths ξ_i to take and the appropriate flight speed V_i so that its own threat exposure $\mathcal{J}_i = f(\xi_i, V_i)$ is minimize and the TOT* value is matched.

6.4 Fault Tolerant Cooperative Rendezvous of UAVs

6.4.1 Motivations

Stringent health and safety and reliability requirements for UAVs have made fault tolerance in UAV control systems quite essential and critical. In presence of undesirable effects such as faults in actuators, control surfaces, or sensors, UAV control systems must autonomously respond and adapt in order to recover from such faults. In particular, for a team of UAVs cooperatively fulfilling a mission, e.g. a rendezvous task, if any of UAVs is subjected to a fault where an individual UAV and/or the supervisory/mission controllers are not equipped with autonomous fault tolerant capabilities, stability of the faulty UAV and success of the mission may not be maintained which could then lead to mission failure.

Motivated from the above, there is an increasing demand for development of autonomous fault diagnosis and reconfigurable UAV cooperative control strategies that are robust to either a fault or gradual performance degradation of individual team members. Consequently, recently fault tolerant control systems have been investigated in multiple UAVs environment (Boskovic et al. [38, 87]). However, most of these work have focused on the formation flying problem. In other words, rendezvous strategies developed thus far in the literature do not formally address health and safety and fault diagnosis issues and solutions.

One of the main contributions of here is incorporation of faults (more specifically, actuator faults) in the UAVs and development of a nonlinear autonomous fault detection and isolation (FDI) scheme for a cooperative rendezvous control problem. Furthermore, a cooperative rendezvous mission re-planning strategy is developed that is initiated as soon as faults are identified in UAVs by our proposed FDI strategy. This will ensure that the rendezvous mission objectives are satisfied, at least partially, even in presence of faults or performance degradations in any of the individual UAVs.

6.4.2 Problem Statement

To facilitate development and demonstration of a fault tolerant cooperative rendezvous control problem of UAVs, a benchmark scenario is formulated by utilizing a specific type of a UAV belonging to the small autonomous unmanned vehicles known as Aerosonde. Through this benchmark, we consider a cooperative rendezvous control problem with three Aerosondes (UAVs) and a single target, where the team of Aerosondes are required to simultaneously arrive at their designated target in presence of *a priori* known threats (distributed over a region of interest (ROI)) despite and even after occurrence of *faults* in actuators of any of the individual Aerosondes. From a strategic and planning perspective, Aerosondes are required to arrive at their designated target simultaneously in order to increase the element of surprise and decrease the risk of being detected by threats and adversaries, thus maximizing the probability of mission success.

The FDI subsystem is tasked with the objective of detecting, isolating, and identifying the severity of the presence of actuator faults that can cause performance degradations in Aerosondes. This will ensure and allow the rendezvous re-planning strategy to accommodate the mission and avoid its failure.

6.4.3 Aerosonde FDI Subsystem

Development of a fault tolerant cooperative rendezvous strategy is a complex problem that requires development of various subsystems and functionalities including an FDI subsystem, cooperative rendezvous algorithms (including a path planning algorithm and an intercept manager), and a re-planning strategy that interfaces the FDI subsystem with the cooperative rendezvous algorithms. Each of these subsystems operate at a different level of abstraction. Thus, we deploy a hierarchical system structure with the FDI subsystem operating at the lowest level (i.e., individual UAV sensors and actuators) and cooperative rendezvous algorithms at the highest, as shown in Figure 6.4. Below, we will first investigate and develop our proposed strategies and solutions for each subsystem separately, and then through numerical simulation results demonstrate and illustrate performance capabilities and advantages of our developed integrated hierarchical fault tolerant cooperative rendezvous scheme, as depicted in Figure 6.4.

Aerosonde (Niculescu [53]) is a small autonomous airplane designed for long-range weather data acquisition. It was developed especially for reconnaissance over oceanic and remote areas, and in harsh conditions. Its low-cost design and operational flexibility has promoted its use for a wide range of remote-sensing applications. In this section, our proposed nonlinear geometric FDI strategy will be applied to the lateral dynamics of the Aerosonde. In this work only the lateral motion of Aerosonde was considered, and which is provided in details

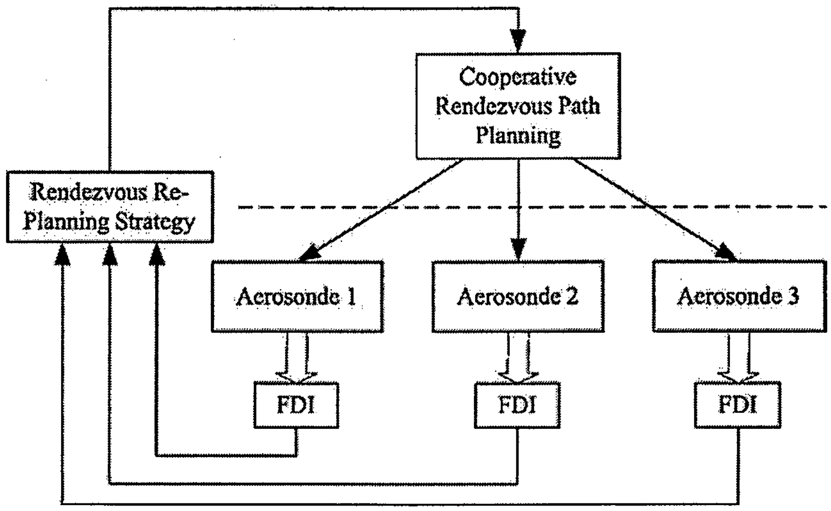

Cooperative Rendezvous Path Planning
Rendezvous Re-Planning Strategy
Aerosonde 1
Aerosonde 2
Aerosonde 3
FDI
FDI
FDI

Figure 6.4: Fault tolerant rendezvous architecture

as follows:

$$\begin{aligned}
m_v \dot{V}_T &= F_T \cos\beta - \frac{1}{2}\rho V_T^2 S\left[C_{D0} + C_D^{\delta_a}\cdot\delta_a + C_D^{\delta_r}\cdot\delta_r\right] \\
m_v \dot{\beta} V_T &= -F_T sin\beta + \frac{1}{2}\rho V_T^2 S\left[C_Y^{\beta}\cdot\beta + C_Y^{\delta_a}\cdot\delta_a + C_Y^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_Y^p\cdot p + C_Y^r\cdot r)\right] - m_v V_T r \\
\Gamma\dot{p} &= J_z\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_l^{\beta}\cdot\beta + C_l^{\delta_a}\cdot\delta_a + C_l^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_l^p\cdot p + C_l^r\cdot r)\right] \\
&\quad + J_{xz}\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_n^{\beta}\cdot\beta + C_n^{\delta_a}\cdot\delta_a + C_n^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_n^p\cdot p + C_n^r\cdot r)\right] \\
\Gamma\dot{r} &= J_{xz}\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_l^{\beta}\cdot\beta + C_l^{\delta_a}\cdot\delta_a + C_l^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_l^p\cdot p + C_l^r\cdot r)\right] \\
&\quad + J_z\cdot\frac{1}{2}\rho V_T^2 Sb\left[C_n^{\beta}\cdot\beta + C_n^{\delta_a}\cdot\delta_a + C_n^{\delta_r}\cdot\delta_r + \frac{b}{2V_T}(C_n^p\cdot p + C_n^r\cdot r)\right] \qquad (6.8)
\end{aligned}$$

where V_T is the airspeed, β is the side slip angle, p is the roll angle rate, r is the yaw angle rate, F_T is the engine thrust, δ_a is the aileron angle, δ_r is the rudder angle, ρ is the air density, S is the wing area, b is the wing span, C_{D0} is the minimum drag, $C_D^{\delta_a}$ and $C_D^{\delta_r}$ are the roll and yaw control derivatives for drag force, respectively, C_Y^{β} is the sideslip derivative, $C_Y^{\delta_a}$ and $C_Y^{\delta_r}$ are the roll and yaw control derivatives for side force, respectively, C_Y^p and C_Y^r are the roll and yaw rate derivatives for side force, respectively, C_l^{β} is the sideslip derivative for roll momentum, $C_l^{\delta_a}$ and $C_l^{\delta_r}$ are the roll and yaw control derivatives for roll momentum, respectively, C_n^p and C_n^r are the roll and yaw rate derivatives for yaw momentum, respectively, C_n^{β} is the sideslip derivative for roll momentum, $C_n^{\delta_a}$ and $C_n^{\delta_r}$ are the roll and yaw control derivatives for yaw momentum, respectively, C_n^p and C_n^r are the roll and yaw rate derivatives for yaw momentum, respectively, J_x, J_y, J_z and J_{xz} are the moment of inertia and m_v is the mass of vehicle.

Consider the state vector as $x = [V_T \ \ \beta \ \ p \ \ r]$, so that the above equations can be written in a compact state space representation

$$\dot{x} = f(x) + g_1(x)F_T + g_2(x)\delta_a + g_3(x)\delta_r \tag{6.9}$$

with appropriate definitions for the functions $f, g_1(x), g_2(x)$, and $g_3(x)$. To address the fault detection and isolation problem in input channels of the above Aerosonde model (equation (6.8)), nonlinear geometric FDI based detection filters are designed (De Persis et al. [49]). Three input channels *aileron* δ_a , *rudder* δ_r *and throttle* F_T are used for controlling the lateral motion of the Aerosondes. Due to dynamics of the Aerosonde, we have shown that (details are omitted due to space limitations) it is not possible to find a residual that is just affected by one channel and is decoupled from other channels. Therefore, a set of three novel detection filters are designed such that each residual is decoupled from one input channel and is affected by two other channels. By applying the nonlinear geometric FDI approach that is modified according to the above observation, the following new set of states are obtained:

$$z_1 = log(x_1) + x_2\frac{C_D^{\delta_a}}{C_Y^{\delta_a}} \tag{6.10}$$

$$z_2 = log(x_1) + x_2\frac{C_D^{\delta_r}}{C_Y^{\delta_r}} \tag{6.11}$$

$$z_3 = p \tag{6.12}$$

$$z_4 = r \tag{6.13}$$

The next step is to design nonlinear observers for z_1, z_2 and z_3 that generate the estimated values of $\hat{z}_1, \hat{z}_2$ and $\hat{z}_3$. Finally, the residual signals are determined according to

$$r_a = z_1 - \hat{z}_1 \tag{6.14}$$

$$r_r = z_2 - \hat{z}_2 \tag{6.15}$$

$$r_F = z_3 - \hat{z}_3 \tag{6.16}$$

Table 6.1 shows our decision logic for determining the detection and isolation of a given fault occurring in one Aerosonde. By observing the FDI residuals and invoking this table, one is able to correctly and reliably detect Aerosonde actuator faults and identify their locations.

Figures 6.5 and 6.6 show the performance of our proposed nonlinear FDI approach. Figure 6.5 depicts the response of the Aerosonde states in a faulty scenario in which a 50% loss of effectiveness engine fault has occurred at 50 seconds. One can clearly observe that after the occurrence of this fault, the Aerosonde is no longer able in achieving the commanded airspeed (23m/s), and where the residuals for aileron and rudder exceed their thresholds as shown in Figure 6.6. Therefore, by using the decision logic as indicated in Table 6.1 and from Figure 6.6, the FDI scheme has successfully identified the occurrence of the engine fault at the instant 50 seconds.

6.4.4 Cooperative Rendezvous Re-planning

In order to facilitate development and implementation of a cooperative rendezvous control and re-planning strategy, we first formally state the general cooperative rendezvous problem

Table 6.1: Fault detection and isolation for Aerosonde UAV

Threshold exceeds	Threshold does not exceed	Detected fault
r_a	r_r, r_F	rudder or engine
r_r	r_a, r_F	aileron or engine
r_F	r_a, r_r	aileron or rudder
r_r, r_F	r_a	aileron
r_a, r_F	r_r	rudder
r_a, r_r	r_F	engine
r_a, r_a, r_F	-	concurrent faults in two channels or more

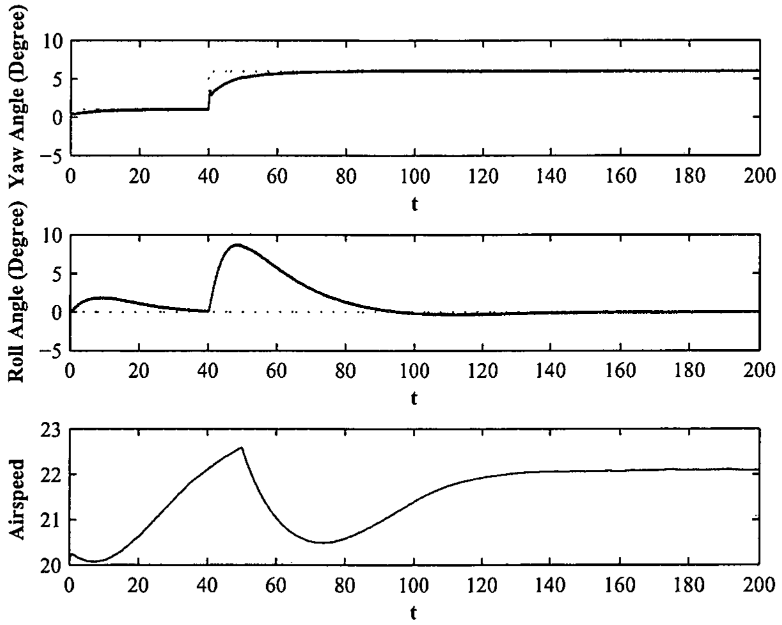


Figure 6.5: Aerosonde states corresponding to a 50% loss of effectiveness fault in the engine

and the steps required for its implementation. For this purpose, consider a group of M unmanned vehicles (UAVs) that are required to traverse through N target locations starting from completely diverse initial positions and in presence of threats in the ROI. The threats can in general be known *a priori*, be "pop-up", or become known when a UAV arrives close to their neighborhood.

In general, the cooperative rendezvous problem can be decomposed into five subproblems (Beard et al. [33]) including (1) target assignment, (2) intercept management, (3) path planning, (4) trajectory generation, and (5) control. However, since we consider a cooperative rendezvous scenario with a *single* target and in presence of *a priori* known threats distributed over the area of interest (AOI), target assignment functionality is not required. In this research, we will mainly focus on intercept management and path planning functionalities and develop a rendezvous re-planning strategy including intercept and path re-planning in the event that faults occur in actuators of individual UAVs.

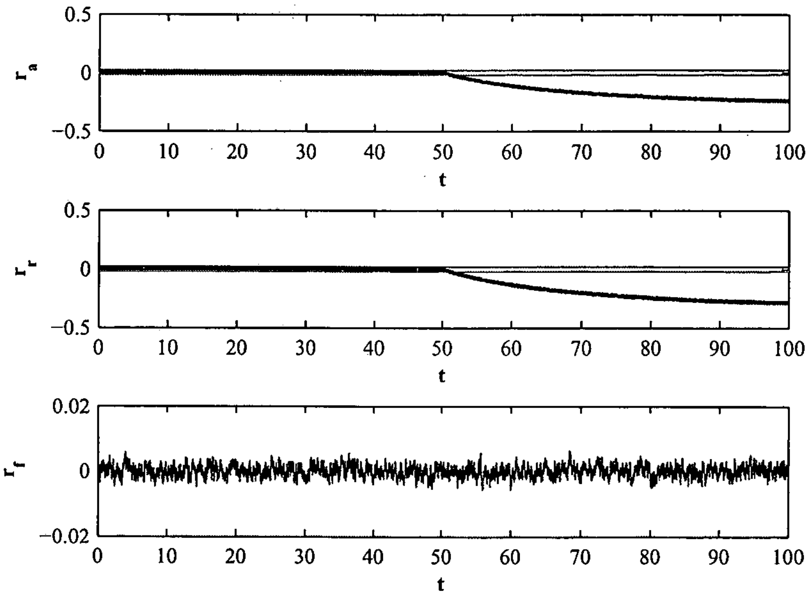


Figure 6.6: Residuals corresponding to a 50% loss of effectiveness fault in the engine

As soon as a fault is detected and isolated in the actuators of any individual Aerosonde, the *severity* of the occurred fault will be calculated and determined. Since loss of effectiveness faults in the actuators are considered, by *severity* we essentially mean the percentage of effectiveness that has been lost in the actuator due to the occurrence of the fault. The fault *severity* value can then be utilized for obtaining the amount of performance degradation in the faulty Aerosonde. Table 6.2 shows the amount of degradation in the maximum achievable speed of Aerosonde for different fault severities. The reason for providing this table for only the Aerosonde speed is that the feasible speed range/limits is a critical and important input to the path planning and intercept manager algorithms, and thus will directly affect the optimal TOT* and paths to the target.

Table 6.2: Fault effects on the Aerosonde speed limits

Fault severity (Loss of effectiveness in engine)	Speed limits of the Aerosonde (m/sec)
0%	18 – 32
10%	18 – 30.77
20%	18 – 29.48
30%	18 – 28.24
40%	18 – 26.65
50%	18 – 24.99
60%	18 – 23.64
70%	18 – 21.54
80%	18 – 19.48

In conclusion, once a fault is detected and isolated and its severity is identified, the re-planning algorithm will determine and obtain new speed constraints using the look-up tables that are given in Table 6.2. Subsequently, the intercept manager and path planning algorithms are executed using the newly obtained constraints on the faulty Aerosonde speed. Therefore, a new optimal TOT* and a new set of optimal paths to the target will then be generated. Table 6.3 shows how loss of effectiveness engine faults affect the optimal TOT*. One can observe that with increasing the severity of faults the team optimal TOT* is also increasing, but not necessarily in a linear fashion.

Table 6.3: Fault effect on optimal TOT*

Loss of effectiveness in engine	0%	10%	20%	30%	40%	50%	60%	70%	80%
Optimal TOT* (sec)	540	561	586	612	648	691	731	802	886

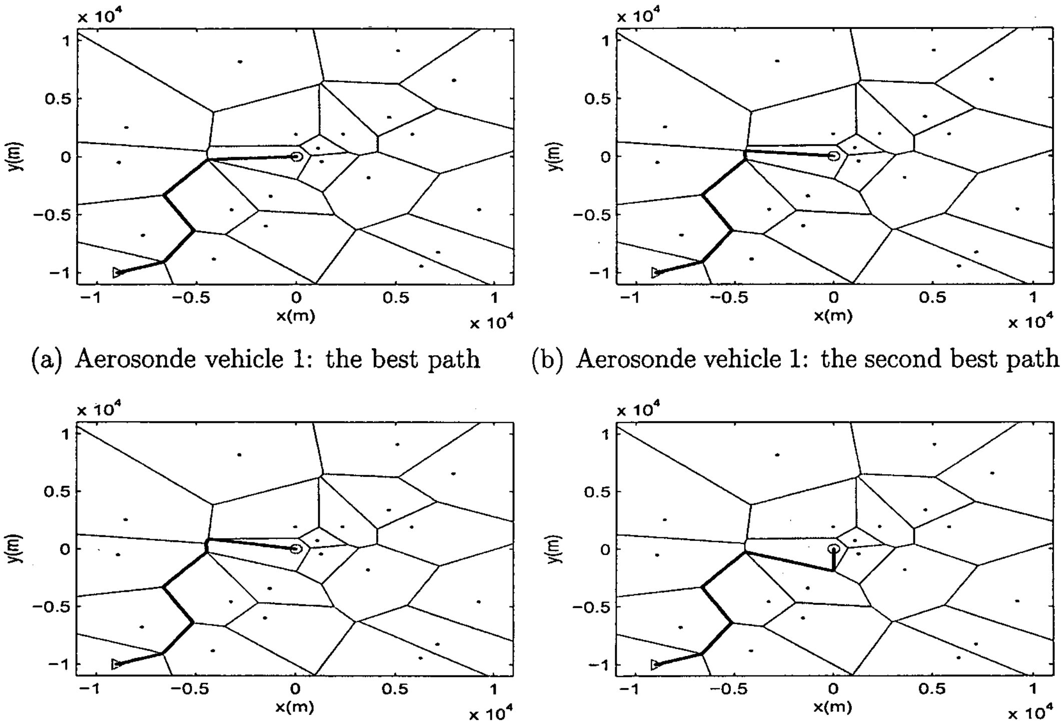
Simulation Results

In this section, simulation results for a cooperative rendezvous path planning and intercept manager are shown. Three simulation scenarios are considered, namely, healthy condition, 50% loss of effectiveness and 80% loss of effectiveness. Figures 6.7, 6.8 and 6.9 depict the 4 best candidate paths for the three Aerosondes in a healthy condition, respectively. These paths are then considered in order to generate coordination functions with nominal speed limits of the Aerosonde (18–32 m/s), which is shown in Figure 6.10. Consequently, the team optimal TOT* is determined, which leads to the cooperative rendezvous paths for all Aerosonde as shown in Figure 6.11. Furthermore, the effects of the faults on rendezvous path planning and intercept manager are investigated. Figures 6.12 and 6.13 depict coordination function of Aerosondes in case of 50% loss of effectiveness faults in engines, and Figures 6.12 and 6.13 show the scenario of 80% loss of effectiveness. One can observe that due to existence of faults, not only the optimal team TOT* changes, but also the optimal paths for Aerosondes need to be changed correspondingly.

6.5 Remarks

In this research, we have developed a fault tolerant cooperative control scheme for UAV rendezvous problem. First, a nonlinear geometric FDI subsystem was developed that can detect and isolate faults in actuators of an unmanned Aerosonde aerial vehicle including engine thrusters (throttle) and control surfaces such as aileron and rudder. A novel fault isolation logic was developed based on different combinations of fault indicators that are known as residuals. The simulation results have shown the effectiveness and utility of our developed FDI scheme.

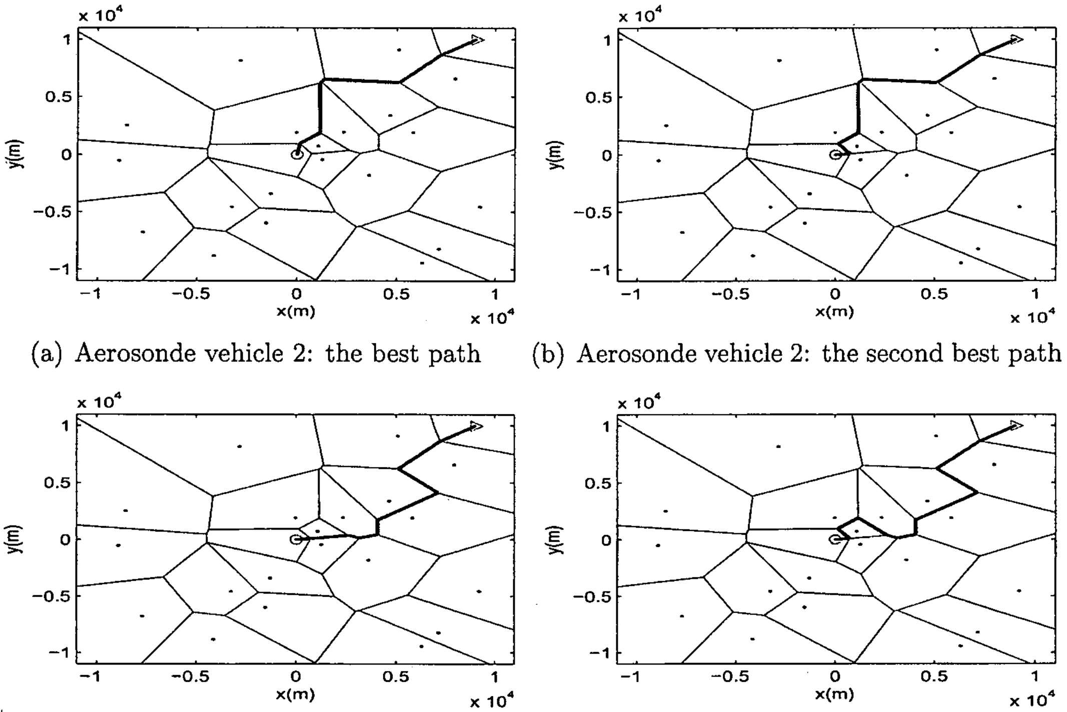
A cooperative rendezvous planning algorithm was also implemented that ensures simultaneous arrival of UAVs at their designated target. Furthermore, a rendezvous re-planning strategy was proposed and developed that provides a cooperative rendezvous algorithm with new constraints on Aerosondes' velocities (resulting from performance degradations due to faults) as soon as faults in their engine thrusters are detected and identified by the FDI



(a) Aerosonde vehicle 1: the best path (b) Aerosonde vehicle 1: the second best path

(c) Aerosonde vehicle 1: the third best path (d) Aerosonde vehicle 1: the fourth best path

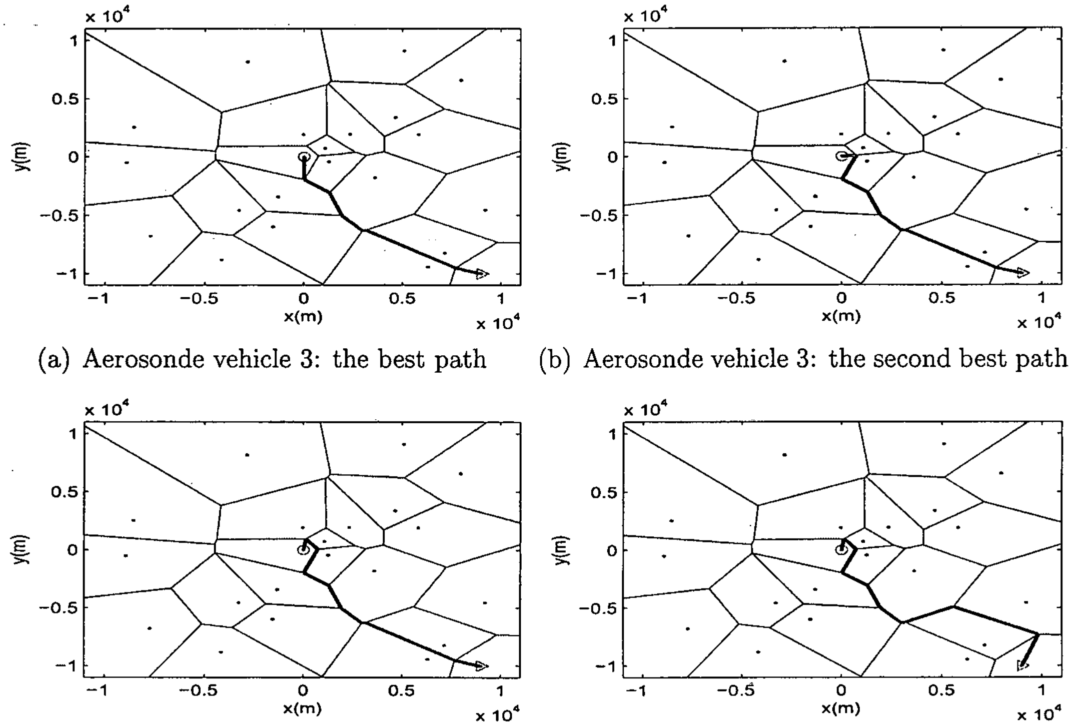
Figure 6.7: The best paths of Aerosonde 1 under healthy condition



(a) Aerosonde vehicle 2: the best path (b) Aerosonde vehicle 2: the second best path

(c) Aerosonde vehicle 2: the third best path (d) Aerosonde vehicle 2: the fourth best path

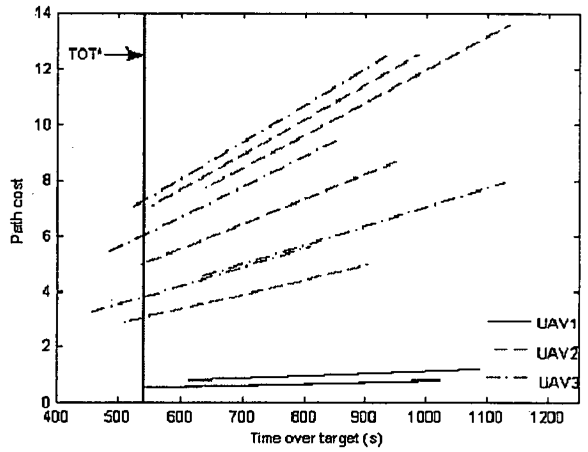
Figure 6.8: The best paths of Aerosonde 2 under healthy condition



(a) Aerosonde vehicle 3: the best path (b) Aerosonde vehicle 3: the second best path

(c) Aerosonde vehicle 3: the third best path (d) Aerosonde vehicle 3: the fourth best path

Figure 6.9: The best paths of Aerosonde 3 udner healthy condition



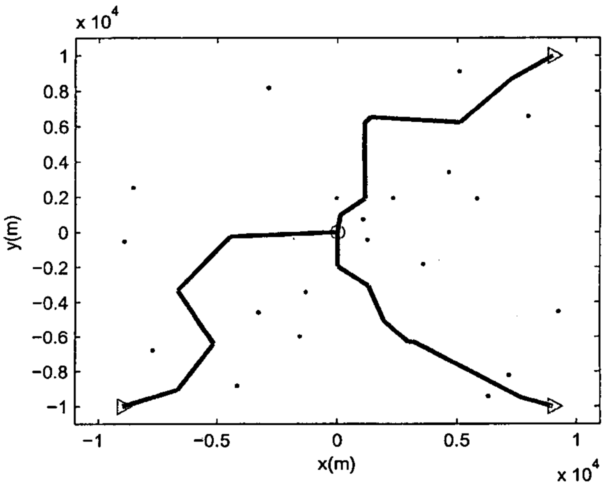
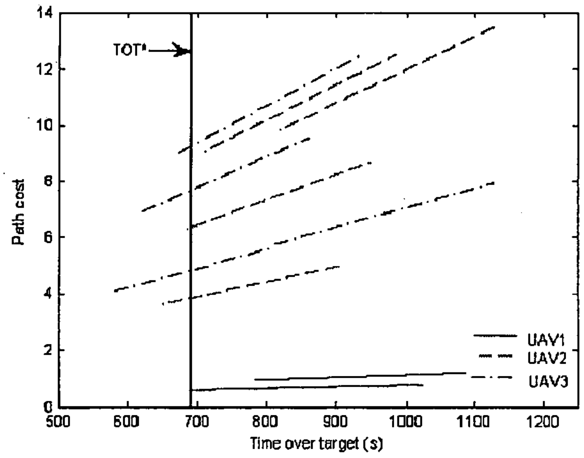


Figure 6.10: Cost functions for all the three healthy Aerosondes

Figure 6.11: Optimal paths for the three healthy Aerosondes with synchronized TOT

subsystem. Consequently, a new set of optimal paths to target and command velocities are generated for each Aerosonde to guarantee UAVs intercept at the designated target despite presence of Aerosonde significant performance degradations due to engine thruster faults. Finally, we have integrated our proposed and developed FDI subsystem, the cooperative rendezvous planning algorithm, and our developed rendezvous re-planning strategy into a



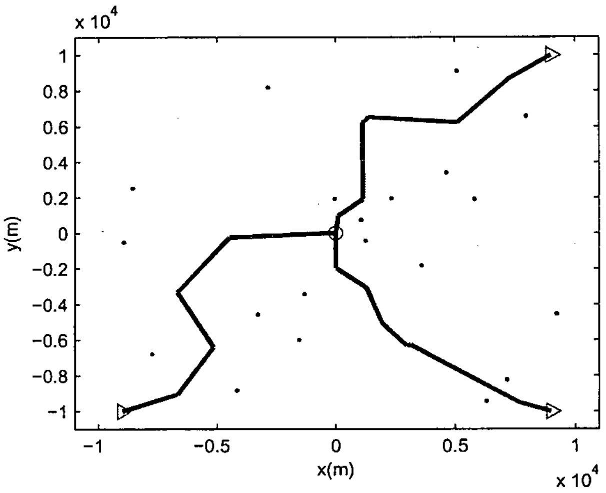
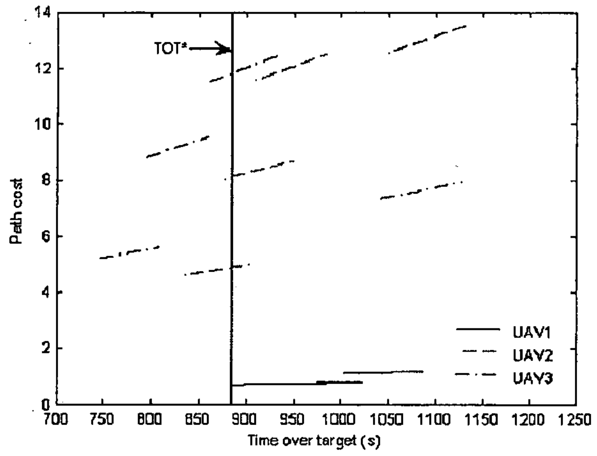


Figure 6.12: Cost functions for all the three Aerosondes when engines lose 50% of effectiveness

Figure 6.13: Optimal paths for the three Aerosondes with synchronized TOT when engines lose 50% of effectiveness



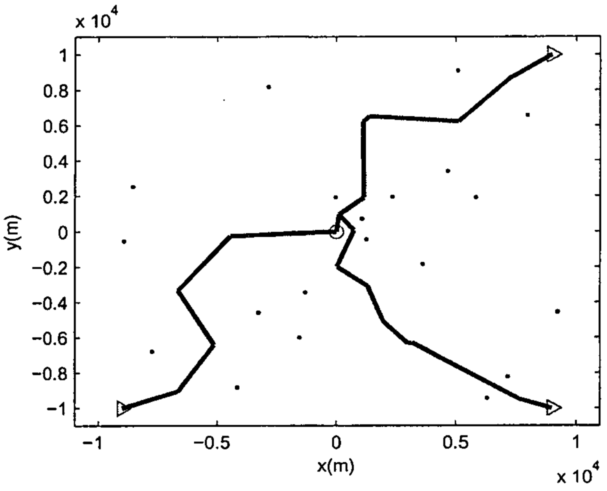


Figure 6.14: Cost functions for all the three Aerosondes when engines lose 80% of effectiveness

Figure 6.15: Optimal paths for the three Aerosondes with synchronized TOT when engines lose 80% of effectiveness

hierarchical fault tolerant cooperative rendezvous scheme. Simulation results have illustrated and demonstrated the effectiveness of our proposed scheme.

Chapter 7

Conclusions and Future Work

In this project, we have solved the problem of fault detection, isolation and recovery for unmanned systems. The developed techniques have been applied to fault diagnosis in three UAV platforms including Almost Lighter Than Air Vehicles (ALTAV), Aerosonde and F16 fixed-wing aircraft. The developed FDI scheme can be broken to two major subsystems. One is the model-based fault detection and isolation subsystem, which has been developed using nonlinear geometric FDI approach and the other is a robustifying module that is implemented by learning-based methodologies, namely neural networks.

Using the nonlinear geometric FDI approach we have been able to detect and identify the occurrence of four types of fault including float, lock in place, loss of effectiveness, and hard over failure in the actuators of the UAV. Furthermore, the proposed scheme is capable of isolating the faulty actuator among the input channels. Moreover, the numerous simulation results have demonstrated the effectiveness of the algorithm in detecting and isolating possible presence of concurrent faults in the system.

Due to the model-based nature of the geometric FDI approach and in order to make the FDI scheme robust to modeling uncertainties, a hybrid framework has been developed that employs neural networks capabilities in learning unknown nonlinear dynamic uncertainties. Using the dynamic neural networks in conjunction with the highly efficient model-based geometric FDI, we have been able to detect, isolate, and identify the nature and the location of the faults in the UAV even in the presence of modeling uncertainties and disturbances.

In order to investigate the FDIR for team of UAVs, we have considered and developed two cooperative strategies. One was the formation flying of ALTAV and the other was the cooperative rendezvous problem for fixed-wing aircraft. Within the formation flying context, we have developed a recovery strategy that leads the faulty ALTAV in the formation to a safe mode of operation. As far as the rendezvous problem is concerned, we have designed and implemented optimal path planning and intercept management algorithms that allow the fixed-wing UAVs to simultaneously arrive at a specified target even in the presence of threats. We have also designed a mission recovery technique for the rendezvous problem. In the proposed solution, as soon as a fault is detected and identified in any of the UAVs in the rendezvous mission, new trajectories are assigned to each UAV based on the newly imposed constraints due to the faults. The new trajectories are calculated using an optimal path replanning strategy.

The results presented in this project have raised many other scientific and technological issues that need to be further investigated and explored in future.

In the context of single UAV fault diagnosis, model-based FDI results need to be extended to the faults in sensors and system components other than actuators. Furthermore, formal analytical derivations need to be explored for the performance characteristics of the proposed robust FDI scheme. Moreover, in order to fully verify the effectiveness of the robustifying scheme, it should be applied to other UAV platforms. The extension of the robustifying scheme to sensor and component faults should also be investigated. To achieve full autonomous fault tolerant capabilities, more advanced adaptive intelligent recovery techniques should be developed that can optimally recover each single UAV from faults in the actuators, sensors and system components.

On the other hand, FDIR research within the mission level of the cooperative control of a team of UAVs is very much in its infancy. The rendezvous mission and formation flying recovery techniques that we have developed in this project need further improvements to satisfy the stringent requirements of a practical UAV cooperative control problem. Furthermore, they should be extended to handle the possible presence of information flow faults between the team members. Generalization of the proposed FDIR techniques to other types of cooperative missions such as reconnaissance and surveillance, to name a few, should also be considered. This will make the FDIR techniques mission independent.

Bibliography

[1] "Capabilities in the area of flight-critical systems." Available on the Web at http://www.ssci.com/Technology/FCS/SofC_nop.pdf.

[2] P. R. Kumar, "New technological vistas for systems and control," in *IEEE Control Systems Magazine*, vol. 21, pp. 24 – 37, 2001.

[3] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control.* New York: John Wiley & Sons, Inc., 3rd ed., 1996.

[4] "http://www.aerosonde.com/,"

[5] R. W. Beard, T. W. Mclain, M. A. Goodrich, and E. P. Anderson, "Coordination variables, coordination functions, and cooperative-timing missions," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 911–922, December 2002.

[6] J. D. Boskovic, S. E. Bergstrom, and R. K. Mehra, "Retrofit reconfigurable flight control in the presence of control effector damage," in *Proceedings of American Control Conference*, vol. 4, pp. 2652 – 2657, 2003.

[7] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, "An overview of emerging results in cooperative uav control," in *Proceedings of 43rd IEEE Conference on Decision and Control*, vol. 1, pp. 602 – 607, 2004.

[8] S. A. Bortoff, "Path planning for uavs," in *Proceedings of American Control Conference*, vol. 20, pp. 34 – 44, 2000.

[9] Y. H. Qu, Q. Pan, and J. G. Yan, "Flight path planning of uav based on heuristically search and genetic algorithms," in *Proceedings of 32nd Annual Conference of IEEE Industrial Electronics Society*, pp. 45 – 49, 2005.

[10] P. Root, J. D. Mot, and E. Feron, "Randomized path planning with deceptive strategies," in *Proceedings of American Control Conference*, vol. 3, pp. 1551 – 1556, 2005.

[11] S. Jeyaraman, A. Tsourdos, r. Zbikowski, and B. White, "Formal techniques for the modeling and validation of a co-operating uav team that uses dubins set for path planning," in *Proceedings of American Control Conference*, vol. 7, pp. 4690 – 4695, 2005.

[12] I. K. Nikolos and A. N. Brintaki, "Coordinated uav path planning using differential evolution," in *Proceedings of IEEE International Symposium on Intelligent Control, Mediterrean Conference on Control and Automation*, pp. 549 – 556, 2004.

[13] T. Shouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of American Control Conference*, vol. 6, pp. 5576 – 5581, 2004.

[14] H. Wong, V. Kapila, and R. Vaidyanathan, "Uav optimal path planning using c-c-c class paths for target touring," in *Proceedings of IEEE Conference on Decision and Control*, vol. 1, pp. 14 – 17, 2004.

[15] M. Jun and R. D'Andrea, *Path planning for unmanned aerial vehicles in uncertain and adversarial environments.* Kluwer, 2002.

[16] P. R. Chandler, S. Rasmussen, and M. Pachter, "Uav cooperative path planning," in *Proceedings of AIAA Guidance, Navigation and Control Conference*, 2000.

[17] T. W. Mclain, P. R. Chandler, S. Rasmussen, and M. Pachter, "Cooperative control of uav rendezvous," in *Proceedings of IEEE American Control Conference*, vol. 3, pp. 2308 – 2314, 2001.

[18] F. Giulieeti, L. Pollini, and M. Innocenti, "Autonomous formation flight," in *IEEE Control Systems Magazine*, vol. 20, pp. 34 – 44, 2000.

[19] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control with applications to multi-vehicle formation stabilization," in *Automatica*, 2004. Submitted to Automatica.

[20] W. Li and C. G. Cassandras, "Stability properties of a receding horizon controller for cooperating uavs," in *Proceedings of 43rd IEEE Conference on Decision and Control*, vol. 3, pp. 2905 – 2910, 2004.

[21] A. Richards and J. How, "Decentralized model predictive control of cooperating uavs," in *Proceedings of 43rd IEEE Conference on Decision and Control*, vol. 4, pp. 4286 – 4291, 2004.

[22] R. Olfati-Saber and R. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proceedings of IEEE Conference on Decision and Control*, 2002.

[23] R. Olfati-Saber and R. Murray, "Flocking with obstacle avoidance: Cooperation with limited information in mobile networks," in *Proceedings of IEEE Conference on Decision and Control*, 2003.

[24] M. Innocenti, G. Mancino, M. Garofoli, and M. Napolitano, "Preliminary analysis of formaiton flight management," in *Proceedings of AIAA Guidance, Navigation and Control Conference*, pp. 258 – 268, 1999.

[25] R. A. Murphey, *Cooperative Control and Optimization*, ch. An Introduction to Collective Systems. Kluwer, 2002.

[26] R. Beard, T. McLain, D. Nelson, and D. Kingston, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," March 2005. Available at https://dspace.byu.edu/handle/1877/60.

[27] D. Casbeer, D. Kingston, R. Beard, T. Mclain, S. M. Li, and R. Mehra, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," December 2001. Available at https://dspace.byu.edu/handle/1877/55.

[28] R. A. Murphey and J. K. O'Neal, "A cooperative testbed architecture for smart loitering weapons," in *Proceedings of the 5th International Conference on Information Fusion*, vol. 1, pp. 694 – 699, 2002.

[29] W. Blake and D. Multhopp, "Design, performance and modeling considerations for close formation flight," in *Proceedings of the AIAA Navigation, Guidance and Control Conference*, pp. 476 – 486, 1998.

[30] C. Schumacher and S. N. Singh, "Nonlinear control of multiple uavs in close-coupled formation flight," in *Proceedings of AIAA Guidance, Navigation, and Control Conference*, 2000.

[31] M. R. Anderson and A. C. Robbins, "Formation flight as cooperative game," in *Proceedings of AIAA Guidance, Navigation and Control Conference*, pp. 244 – 251, 1998.

[32] J. A. Fax and R. M. Murray, "Graph laplacians and stabilization of vehicle formations," in *Proceedings of IFAC World Congress*, vol. 1, 2002.

[33] R. Beard, T. McLain, M. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911 – 922, 2002.

[34] M. Flint, M. Polycarpou, and E. F. Gaucherand, "Cooperative control for multiple autonomous uav's searching for targets," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 2823 – 2828, 2002.

[35] M. Flint, E. F. Gaucherand, and M. Polycarpou, "Cooperative control for uav's searching risky environments for targets," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 3567 – 3572, 2003.

[36] Y. Yang, A. Minai, and M. Polycarpou, "Decentralized cooperative search by networked uavs in an uncertain environment," in *Proceedings of the American Control Conference*, pp. 5558 – 5563, 2004.

[37] Y. Jin, Y. Liao, M. Polycarpou, and A. Minai, "Balancing search and target response in cooperative uav teams," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 2923 – 2928, 2004.

[38] J. D. Boskovic and R. K. Mehra, "Stable adaptive multiple model-based control design for accommodate on of sensor failures," in *Proceedings of American Control Conference*, pp. 2046 – 2051, 2002.

[39] J. Brinker and K. Wise, "Reconfigurable flight control of a tailless advance fighter aircraft," in *Proceedings of the 1998 AIAA Guidance, Navigation and Control Conference*, vol. 1, pp. 75 – 87, 1998.

[40] V. Liberatore, W. S. Newman, and K. Bhasin, "Ip communication and distributed agents for unmanned autonomous vehicles," 2003.

[41] F. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks," in *IEEE Control Systems Magazine*, vol. 21, pp. 66 – 83, 2001.

[42] Z. Wu, H. Kumar, and A. Davari, "Performance evaluation of ofdm transmission in uav wireless communication," in *Proceedings of the 37th Southeastern Symposium on System Theory*, pp. 6 – 10, 2005.

[43] T. S. Rappaport, *Wireless Communications.* Upper Saddle River, NJ, Prentice Hall, 2002.

[44] R. W. Beard and T. W. McLain, "Multiple uav cooperative search under collision avoidance and limited range communication constraints," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 25 – 30, 2003.

[45] Y. Liao, Y. Jin, A. Minai, and M. Polycarpou, "Information sharing in cooperative unmanned aerial vehicle teams," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pp. 90 – 95, 2005.

[46] J. Lou, "Some new optimal control problems in uav cooperative control with information flow constraints," in *Proceedings of the American Control Conference*, pp. 2181 – 2186, 2003.

[47] L. Pollini, F. Giulietti, and M. Innocenti, "Robustness to communication failures within formation flight," in *Proceedings of the American Control Conference*, pp. 2860 – 2866, 2002.

[48] R. K. Mehra, J. D. Boskovic, and S. M. Li, "Autonomous formation flying of multiple ucavs under communication failure," in *Proceedings of the IEEE Position Location and Navigation Symposium*, pp. 371 – 378, 2000.

[49] C. D. Persis and A. Isidori, "A geometric approach to nonlinear fault detection an isolation," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 853 – 865, 2001.

[50] A. Isidori, *Nonlinear Control Systems.* New York: Springer Verlag, 3rd ed., 1995.

[51] G. B. Folland, *Real Analysis: Model Techniques and Their Applications.* John Wiley & Sons, Inc., 2nd ed., 1999.

[52] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation.* Hoboken, NJ: John Wiley & Sons, Inc., 2nd ed., 2003.

[53] M. Niculescu, "Lateral track law for aerosonde uav," *39th AIAA Aerospace Sciences Meeting and Exhibit*, 2001-0016.

[54] "Aerosim aeronautical simulation blockset," *Umannaed Dynamics.* www.u_dynamics.com.

[55] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems.* Kluwer, 1999.

[56] K. Watanabe and D. M. Himmelblau, "Instrument fault detection in systems with uncertainties," *Int. J. Sys. Sci.*, vol. 13, no. 2, pp. 137–158, 1982.

[57] R. J. Patton, S. W. Willcox, and S. J. Winter, "A parameter insensitive technique for aircraft sensor fault analysis," *J. Guidance, Control, and Dynamics*, vol. 10, no. 3, pp. 359–367, 1987.

[58] X. Lou, A. S. Willsky, and G. C. Verghese, "Optimally robust redundancy relations for failure detection in uncertain systems," *Automatica*, vol. 22, no. 3, pp. 333–344, 1986.

[59] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[60] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, vol. 1, pp. 4–27, March 1990.

[61] K. S. Narendra and K. Parthasarathy, "Neural networks for control theory and practice," *Proceedings of the IEEE*, vol. 84, pp. 1385–1406, Oct. 1996.

[62] T. Parisini and R. Zoppoli, "Neural networks for nonlinear state estimation," *Int. J. of Robust and Nonlinear Control*, vol. 4, no. 2, pp. 231–248, 1994.

[63] A. Alessandri, M. Baglietto, T. Parisini, and R. Zoppoli, "A neural state estimator with bounded errors for nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 44, pp. 2028–2042, Nov. 1999.

[64] P. M. Frank and B. Koppen-Seliger, "Fuzzy logic and neural network applications to fault diagnosis," *Int. J. of Approximate Reasoning*, vol. 16, no. 1, pp. 67–88, 1997.

[65] C. Lopez-Toribio, R. Patton, and F. Upal, "Artificial intelligence approaches to fault diagnosis for dynamic systems," *Int. J. of Appl. Mathematics and Comp. Science*, vol. 9, no. 3, 1999.

[66] T. Marcu, L. Mirea, and P. Frank, "Development of dynamic neural networks with application to observer-based fault detection and isolation," *Int. J. of Appl. Mathematics and Comp. Science*, vol. 9, no. 3, 1999.

[67] R. Patton, F. Upal, and C. Lopez-Toribio, "Soft computing approaches to fault diagnosis for dynamic systems: A survey," in *Proceedings of 4th IFAC Symposium on Fault Detection, Supervision and Safety for Tech. Processes, Budapest*, pp. 298–311, 2000.

[68] A. T. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Trans. on Neural Networks*, vol. 8, pp. 1410–1420, November 1997.

[69] A. B. Trunov and M. M. Polycarpou, "Robust fault diagnosis of state and sensor faults in nonlinear multivariable systems," in *Proceedings of the American Control Conference, San Diego, California*, pp. 608 — 612, 1999.

[70] X. Zhang, M. M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. on Neural Networks*, vol. 47, pp. 576–593, April 2002.

[71] A. Yazdizadeh, *Idenfication of Nonlinear Systems Using Dynamic Neural Networks.* PhD thesis, Department of Electrical and Computer Engineering, Concordia University, 1997.

[72] R. H. Brown, T. L. Buchti, and X. Feng, "Artificial neural network identification of partially known dynamic nonlinear systems," in *Proceedings of the 32nd Conference on Decision and Control*, pp. 3694 – 3699, 1993.

[73] k. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, pp. 801 – 806, 1993.

[74] C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 144 – 156, 1995.

[75] A. Parlos, A. Atiya, and K. Chong, "Recurrent multilayer perceptron for nonlinear system identification," in *Proceedings of International Joint Conference on Neural Networks*, pp. II–537–540, 1991.

[76] A. Atiya and A. Parlos, "Nonlinear system identification using spatiotemporal neural networks," pp. II–504–509, 1992.

[77] J. C. Principe and M. A. Motter, "System identification with dynamic neural networks," in *Proceedings of the World Congress on Neural Networks*, pp. II–284–289, 1994.

[78] D. H. Rao, M. M. Gupta, and H. C. Wood, "Adaptive inverse control of nonlinear systems using dynamic neural networks," in *Proceedings of the World Congress on Neural Networks*, pp. II–305–310, 1994.

[79] A. Yazdizadeh and K. Khorasani, "Nonlinear system identification using adaptive time delay neural networks," in *Proceedings of the World Congress on Neural Networks*, pp. 482 – 485, 1996.

[80] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 37, no. 3, pp. 328 – 339, 1989.

[81] F. Abdollahi, H. A. Talebi, and R. V. Patel, "A stable neural network-based identification scheme for nonlinear systems," in *Proceedings of American Control Conference*, vol. 4, pp. 3590–3595, 2003.

[82] F. Abdollahi, H. A. Talebi, and R. V. Patel, "Stable identification of nonlinear systems using neural networks: theory and experiments," *IEEE/ASME Trans. on Mechatronics*, vol. 11, pp. 488–495, August 2006.

[83] T. McLain and R. Beard, "Trajectory planning for coordinated rendezvous of unmanned air vehicles," in *Proceedings of AIAA Guidance, Navigation and Control Conference*, 2000.

[84] T. W. McLain, P. R. Chandler, S. Rasmussen, and M. Pachter, "Cooperative control of uav rendezvous," in *Proceedings of the American Control Conference*, pp. 2309–2314, June 2001.

[85] T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative-timing missions," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 150–161, January-Feburay 2005.

[86] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications.* New York: Springer-Verlag, 2000.

[87] J. D. Boskovic and R. K. Mehra, "An adaptive reconfigurable formation flight control design," in *Proceedings of American Control Conference*, pp. 284–289, 2003.

DISTRIBUTION LIST

Internal

Document Library (2)

External

DRDKIM (unbound copy)

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(Highest Classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

1. ORIGINATOR (name and address)
Department of Electrical and Computer Engineering
Concordia University, S-EV005.139
1515 St.Catherine West
Montreal, Quebec H4G 2W1 Canada

2. SECURITY CLASSIFICATION
(Including special warning terms if applicable)
UNCLASSIFIED

3. TITLE (Its classification should be indicated by the appropriate abbreviation (S, C, R or U)
Nonlinear Fault Detection, Isolation and Recovery Techniques for Unmanned Systems

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)
K. Khorasani

5. DATE OF PUBLICATION (month and year) March 30, 2007	6a. NO. OF PAGES 230	6b .NO. OF REFERENCES 87

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. Give the inclusive dates when a specific reporting period is covered.)
Contractor Report

8. SPONSORING ACTIVITY (name and address)
DRDC Valcartier

9a. PROJECT OR GRANT NO. (Please specify whether project or grant) 13jz03	9b. CONTRACT NO. W7701-052053/001/QCL
10a. ORIGINATOR'S DOCUMENT NUMBER QCB-5-20386	10b. OTHER DOCUMENT NOS N/A

AC1052
PP 50 A5

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

☒ Unlimited distribution
□ Restricted to contractors in approved countries (specify)
□ Restricted to Canadian contractors (with need-to-know)
□ Restricted to Government (with need-to-know)
□ Restricted to Defense departments
□

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)
None

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(Highest Classification of Title, Abstract, Keywords)

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

In order to avoid adverse consequences due to failures, it is desirable to have an advanced failure detection and isolation (FDI) system that detects and identifies anomalies early to minimize the damage, and that can remedy as many failures as possible. In complex systems, fault diagnosis is typically accomplished using a hierarchical approach. In our proposed autonomous unmanned vehicle (UAV) system, fault diagnosis, isolation and recovery (FDIR) is accomplished by using a hierarchical and decentralized approach. At this level of the hierarchy the model based or analytical redundancy based approach to FDIR would require a mathematical model of the process or sub-process under consideration. Based on this knowledge quantities called residuals will be generated. The residuals should be small or close to zero when there are no failure in the system. On the other hand, they should become nonzero and grow large if there are malfunctions in the system. This will accomplish the failure detection. The next important task will be the design of a fault isolation module that would isolate the faulty components or subsystems. There are two major approaches to the design and implementation of recovery procedures, One is to synthesize the procedures for every possible failure mode at the design stage. One the diagnostic and recovery system is activated, it monitors the system and if it detects a failure, then the system will initiate the appropriate recovery procedure. In the other approach, suitable recovery procedures are generated "on-line" upon the detection of failures. In this report, we will examine the advantages and drawbacks of the above approaches in our framework.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

UAV

Fault detection, Isolation, and Recovery

Cooperative Control